

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**



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.....	45

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 ⁹)	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.

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-10:2014

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.

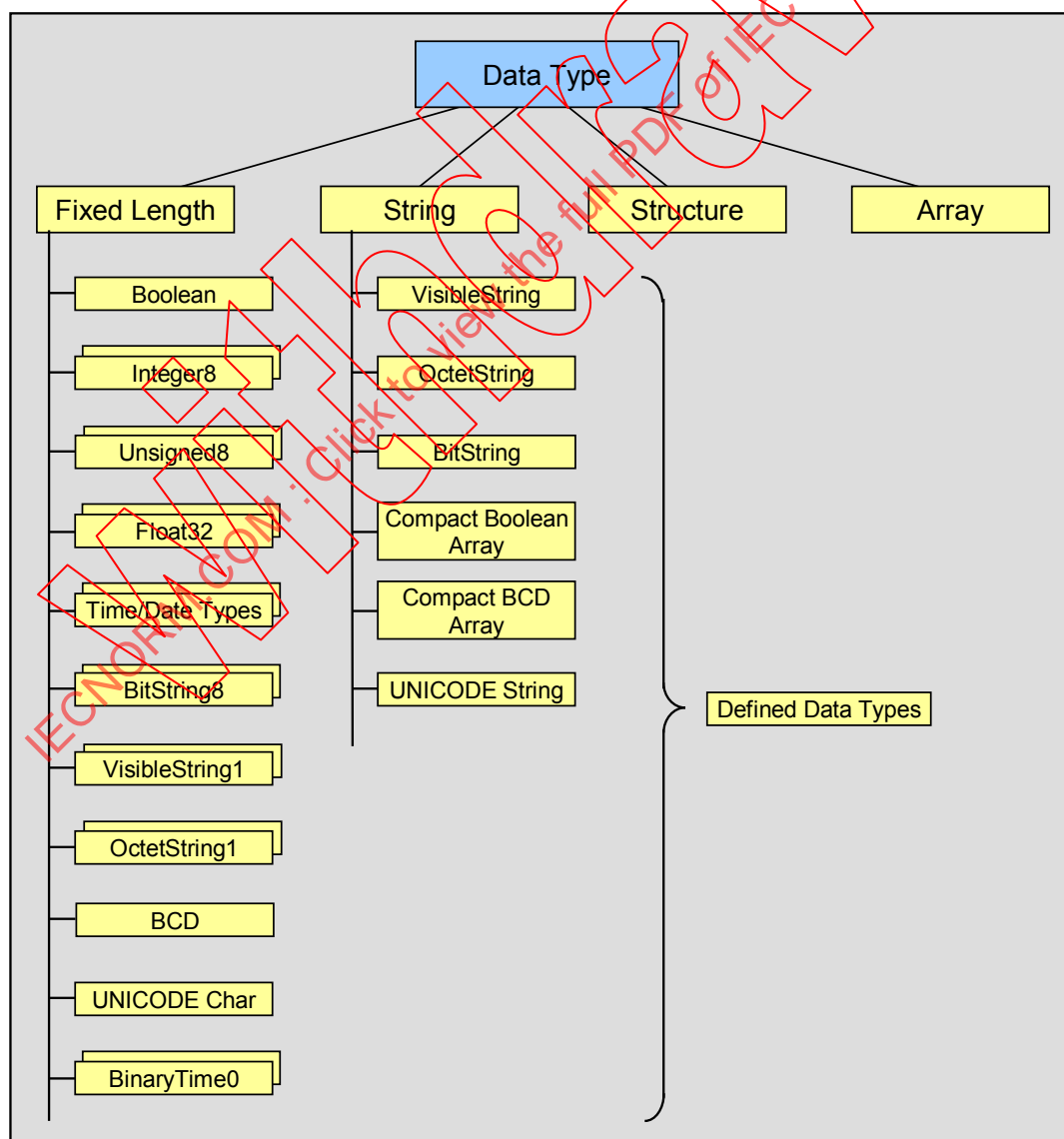


Figure 1 – Data type class hierarchy example

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					

Bit	8	7	6	5	4	3	2	1
Octet 2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}
	fraction							

Table 11 – Unipolar2.16 value range

Range of values	Resolution
$0 \% \leq i \leq (400 \cdot 2^{-14}) \%$	$2^{-14} = 0,0061 \%$

NOTE This type is expressed in IEC 60870 as 'unsigned fixed point number' and used in IEC 61375-1, too.

5.3.1.5.3 Integer types

5.3.1.5.3.1 Integer8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 2
2	Data type Name	= Integer8
3	Format	= FIXED LENGTH
4.1	Octet Length	= 1

This integer type is a two's complement binary number with a length of one octet.

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

5.3.1.5.3.2 Integer16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 3
2	Data type Name	= Integer16
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This integer type is a two's complement binary number with a length of two octets.

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

5.3.1.5.3.3 Integer32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 4
2	Data type Name	= Integer32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This integer type is a two's complement binary number with a length of four octets.

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

5.3.1.5.3.4 Integer64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 55
2	Data type Name	= Integer64
3	Format	= FIXED LENGTH

4.1 Octet Length = 8

This integer type is a two's complement binary number with a length of eight octets.

5.3.1.5.3.5 Normalized value N2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 113
2	Data type Name	= N2
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This integer type is the same as Integer16, i.e. a two's complement binary number with a length of two octets; however having a special interpretation.

The value provides a linear normalized value, such that the integer value 2^{14} (i.e. 16 384) equates to 100 %. The value is thus interpreted according to the following formula:

$$\text{Normalized Value N2 (in \%)} = \text{Value} / 16\,384 \quad (1)$$

where

Normalized Value N2 is the calculated linear normalized value
Value is the raw value for interpretation

The data type offers thus a resolution of 0,0061 % ($= 1 / 16\,384$). The N2 value range is shown in Table 12.

Table 12 – N2 value range

Value	Meaning
-32 768	-200 %
-32 767 ... -1	-199,993 896 484 375 % ... -0,006 1 %
-16 384	-100 %
0	0 %
16 384	100 %
1 ... 32 766	0,006 1 % ... 199,987 792 968 75 %
32 767	199,993 896 484 375 %

5.3.1.5.3.6 Normalized value N4

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 114
2	Data type Name	= N4
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This integer type is the same as Integer32, i.e. a two's complement binary number with a length of four octets; however having a special interpretation.

The value provides a linear normalized value, such that the integer value 2^{30} (i.e. 1 073 741 824) equates to 100 %. The value is thus interpreted according to the following formula:

$$\text{Normalized Value N4 (in \%)} = \text{Value} / 1\,073\,741\,824 \quad (2)$$

where

Normalized Value N4 is the calculated linear normalized value

Value is the raw value for interpretation

The data type offers thus a resolution of approx. 0,000 000 093 1 % (= 1 / 1 073 741 824). The N4 value range is shown in Table 13.

Table 13 – N4 value range

Value	Meaning
-2 147 483 648	-200 %
-1 073 741 824	-100 %
-2 147 483 647 .. -1	approx. -199,999 999 906 9 % ... -0,000 000 093 1 %
0	0 %
1 .. 2 147 483 646	approx. 0,000 000 093 1 % ... 199,999 999 813 7 %
1 073 741 824	100 %
2 147 483 647	approx. 199,999 999 906 9 %

5.3.1.5.3.7 Variable normalized X2

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 123
2	Data type Name	= X2
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This integer type is the same as Integer16, i.e. a two's complement binary number with a length of two octets, however having a special interpretation.

The value is interpreted as a linear normalized value (a value expressed as percentage). The interpretation is similar to the data type N2, however the normalization to 100 % is variable and determined by an additional parameter "X" outside the scope of this data type, such that the integer value 2^X equates to 100 %.

NOTE 1 If the additional specified constant X equates to 14, the value range of X2 equates to that of the data type N2.

The value is interpreted according to the following formula:

$$\text{Normalized Value X2 (in \%)} = \text{Value} / 2^X \quad (3)$$

where

Normalized Value X2 is the calculated linear normalized value

Value is the raw value for interpretation

X is the exponent for calculation, outside the scope of datatype definition

The data type offers thus a resolution of $1 / 2^X$ % or else 2^{-X} . Lower and upper bound of the value are computed according to the following formulae:

$$\text{Lower Bound X2 (in \%)} = -2^{(15-X)} \quad (4)$$

$$\text{Upper Bound X2 (in \%)} = 2^{(15-X)} - 2^{-X} \quad (5)$$

where

Lower Bound X2 is the lower bound of the calculated value
Upper Bound X2 is the upper bound of the calculated value
X is the exponent for calculation, outside the scope of datatype definition

The X2 value range for selected values of “X” is shown in Table 14.

Table 14 – X2 value range

Parameter “X”	Value	Meaning
—	-32 768 ... 32 767	$-2^{(15-X)} \% \leq i \leq (2^{(15-X)} - 2^{-X}) \%$
e.g. X = 10	-32 768 ... 32 767	$-3\,200 \% \leq i \leq -3\,199,90 \%$
e.g. X = 12	-32 768 ... 32 767	$-800 \% \leq i \leq 799,98 \%$
e.g. X = 14	-32 768 ... 32 767	$-200 \% \leq i \leq 199,99 \%$

NOTE 2 The value of this normalized value parameter refers to the additional specified constant X. The associated constant X is required in order to interpret the internal value.

5.3.1.5.3.8 Variable normalized X4

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 124
2	Data type Name	= X4
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This integer type is the same as Integer32, i.e. a two's complement binary number with a length of four octets; however having a special interpretation.

The value is interpreted a linear normalized value (a value expressed as percentage). The interpretation is similar to the data type N4, however the normalization to 100 % is variable and determined by an additional parameter “X” outside the scope of this data type, such that the integer value 2^X equates to 100 %.

NOTE 1 If the additional specified constant X equates to 30, the value range of X4 equates to that of the data type N4.

The value is interpreted according to the following formula:

$$\text{Normalized Value X4 (in \%)} = \text{Value} / 2^X \quad (6)$$

where

Normalized Value X4 is the calculated linear normalized value
Value is the raw value for interpretation
X is the exponent for calculation, outside the scope of datatype definition

The data type offers thus a resolution of $1 / 2^X \%$ or else 2^{-X} . Lower and upper bound of the value are computed according to the following formulae:

$$\text{Lower Bound } X4 \text{ (in \%)} = -2^{(31-X)} \quad (7)$$

$$\text{Upper Bound } X4 \text{ (in \%)} = 2^{(31-X)} - 2^{-X} \quad (8)$$

where

Lower Bound X4 is the lower bound of the calculated value

Upper Bound X4 is the upper bound of the calculated value

X is the exponent for calculation, outside the scope of datatype definition

The X4 value range for selected values of “X” is shown in Table 15.

Table 15 – X4 value range

Parameter “X”	Value	Meaning
—	–2 147 483 648 ... 2 147 483 647	$-2^{(31-X)} \% \leq i \leq (2^{(31-X)} - 2^{-X}) \%$
e.g. X = 26	–2 147 483 648 ... 2 147 483 647	$-3\,200 \% \leq i \leq 3\,200 \%$
e.g. X = 28	–2 147 483 648 ... 2 147 483 647	$-800 \% \leq i \leq 800 \%$
e.g. X = 30	–2 147 483 648 ... 2 147 483 647	$-200 \% \leq i \leq 200 \%$

NOTE 2 The value of this normalized value parameter refers to the additional specified constant X. The associated constant X is required in order to interpret the internal value.

5.3.1.5.3.9 Fixed point value C4

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 122
- 2 Data type Name = C4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

This data type is the same as Integer32, i.e. a two's complement binary number with a length of four octets; however having a special interpretation.

The value is interpreted a linear fixed point value with four decimal places according to the following formula:

$$\text{Fixed Point Value C4} = \text{Value} / 10\,000 \quad (9)$$

where

Fixed Point Value C4 is the calculated linear fixed point value

Value is the raw value for interpretation

The data type offers thus a resolution of 0,000 1 (= 1 / 10 000). The C4 value range is shown in Table 16.

Table 16 – C4 value range

Value	Meaning
–2 147 483 648	–214 748,3648
0	0
2 147 483 647	214 748,3647

5.3.1.5.4 Unsigned types

5.3.1.5.4.1 Unsigned8

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 5
2	Data type Name	= Unsigned8
3	Format	= FIXED LENGTH
4.1	Octet Length	= 1

This type is a binary number. The most significant bit of the most significant byte is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of one octet.

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

5.3.1.5.4.2 Unsigned16

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 6
2	Data type Name	= Unsigned16
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This type is a binary number. The most significant bit of the most significant byte is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets.

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

5.3.1.5.4.3 Unsigned32

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 7
2	Data type Name	= Unsigned32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This type is a binary number. The most significant bit of the most significant byte is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of four octets.

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

5.3.1.5.4.4 Unsigned64

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 56
2	Data type Name	= Unsigned64
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

This type is a binary number. The most significant bit of the most significant byte is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of eight octets.

5.3.1.5.5 Time constant T2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	118
2	Data type Name	=	T2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This integer type is the same as Unsigned16, i.e. a two's complement binary number with a length of two octets; however having a special interpretation.

The value provides a time value based on a defined sampling time "Ta" outside the scope of this data type.

The value is interpreted according to the following formula:

$$\text{Time Value} = \text{Value} * T_a \quad (10)$$

where

<i>Time Value</i>	is the calculated time value
<i>Value</i>	is the raw value for interpretation
<i>T_a</i>	is the defined sampling time, outside the scope of this data type

The data type offers thus a resolution of "Ta". The T2 value range is shown in Table 17. Note that the value range is restricted; values outside the result provide a time value of "0".

Table 17 – T2 value range

Value	Meaning
0 – 32 767	Value * Ta
32 768 – 65 535	0

NOTE The value of this time parameter refers to the additional specified constant sampling time Ta. The associated sampling time Ta is required in order to interpret the internal value.

5.3.1.5.6 Time constant T4

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	119
2	Data type Name	=	T4
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This integer type is the same as Unsigned32, i.e. a two's complement binary number with a length of four octets; however having a special interpretation.

The value provides a time value based on a defined sampling time "Ta" outside the scope of this data type.

The value is interpreted according to the following formula:

$$\text{Time Value} = \text{Value} * T_a \quad (11)$$

where

<i>Time Value</i>	is the calculated time value
<i>Value</i>	is the raw value for interpretation
<i>T_a</i>	is the defined sampling time, outside the scope of this data type

The data type offers thus a resolution of “Ta”. The T4 value range is shown in Table 18. In contrary to the data type T2 the value range is not restricted.

Table 18 – T2 value range

Value	Meaning
0 – 4 294 967 295	Value * Ta

NOTE The value of this time parameter refers to the additional specified constant sampling time Ta. The associated sampling time Ta is required in order to interpret the internal value.

5.3.1.5.7 Time constant D2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 120
2	Data type Name	= D2
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

This integer type is the same as Unsigned16, i.e. a two's complement binary number with a length of two octets; however having a special interpretation.

The value provides a time value based on a defined sampling time “Ta” outside the scope of this data type.

The value is interpreted according to the following formula:

$$\text{Time value} = \text{Value} \times (\text{Ta} / 16\,384) \quad (12)$$

where

<i>Time Value</i>	is the calculated time value
<i>Value</i>	is the raw value for interpretation
<i>T_a</i>	is the defined sampling time, outside the scope of this data type

The data type offers thus a fraction of the constant sampling time with a resolution of “Ta / 16384”. The D2 value range is shown in Table 19. Note that the value range is restricted; values outside the result provide a time value of “0”.

Table 19 – D2 value range

Value	Meaning
0 – 32 767	Value * (Ta / 16 384)
32 768 – 65 535	0

NOTE The value of this time parameter refers to the additional specified constant sampling time Ta. The associated sampling time Ta is required in order to interpret the internal value.

5.3.1.5.8 Time constant R2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	117
2	Data type Name	=	R2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This integer type is the same as Unsigned16, i.e. a two's complement binary number with a length of two octets; however having a special interpretation.

The value provides a time value based on a defined sampling time “Ta” outside the scope of this data type.

The value is interpreted according to the following formula:

$$\text{Time value} = 16\,384 * (\text{Ta} / \text{Value}) \quad (13)$$

where

<i>Time Value</i>	is the calculated time value
<i>T_a</i>	is the defined sampling time, outside the scope of datatype definition
<i>Value</i>	is the raw value for interpretation

The data type offers thus a reciprocal multiple of the constant sampling time with a resolution of “Ta / Value”. The R2 value range is shown in Table 20. Note that the value range is restricted; values outside the result provide a time value of “0”.

Table 20 – R2 value range

Value	Meaning
0	16 384
1 – 16 384	16 384 * (Value / Ta)
16 385 – 65 535	16 384

NOTE The value of this time parameter refers to the additional specified constant sampling time Ta. The associated sampling time Ta is required in order to interpret the internal value.

5.3.1.6 OctetString character types

5.3.1.6.1 UUID

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	1 025
2	Data type Name	=	UUID
3	Format	=	STRUCTURE
5.1	Number of Fields	=	4
5.2.1	Field Name	=	Data1
5.2.2	Field Data type	=	Unsigned32
5.2.3	Field Name	=	Data2
5.2.4	Field Data type	=	Unsigned16
5.2.5	Field Name	=	Data3
5.2.6	Field Data type	=	Unsigned16
5.2.7	Field Name	=	Data4

- 5.2.8.1 Format = ARRAY
 5.2.8.4.1 Number of Array Elements = 8
 5.2.8.4.2 Array Element Data type = Unsigned8

This data type defines a 16 octet fixed length data type. The semantic is specified by The Open Group – Publication C706 and beyond the scope of this standard.

This data type is structured as follows.

Data1

This field contains the first eight hexadecimal digits of the Universally Unique Identifier (UUID).

Data2

This field contains the first group of four hexadecimal digits of the UUID.

Data3

This field contains the second group of four hexadecimal digits of the UUID.

Data4

This field contains an array of eight elements. The first two elements specify the third group of four hexadecimal digits of the UUID. The remaining six elements specify the final 12 hexadecimal digits of the UUID.

Specific values for decentralized peripherals items are shown in Table 21.

Table 21 – UUID for decentralized peripherals

Value	Description
UUID_NIL 00000000-0000-0000-0000-000000000000	Default AR in a device
UUID_IO_ObjectInstance_XYZ DEA00000-6C97-11D1-8271-{xxxxyyyyzzzz}	Identifies a special object instance within a physical device in case there are more than one where - xxxx represents the instance or node number - yyyy identify the Device ID as a vendor specific number for the device class and - zzzz represents the Vendor ID as a central administrative number.
UUID_IO_DeviceInterface DEA00001-6C97-11D1-8271-00A02442DF7D	Identifies the IO device interface uniquely.
UUID_IO_ControllerInterface DEA00002-6C97-11D1-8271-00A02442DF7D	Identifies the IO controller interface uniquely.
UUID_IO_SupervisorInterface DEA00003-6C97-11D1-8271-00A02442DF7D	Identifies the IO supervisor interface uniquely.
UUID_IO_ParameterServerInterface DEA00004-6C97-11D1-8271-00A02442DF7D	Identifies the IO parameter server interface uniquely.

5.3.1.7 Time types

5.3.1.7.1 General

The limitation of 5.1.5 applies.

5.3.1.7.2 NetworkTime

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 58
- 2 Data type Name = NetworkTime
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 8

This data type is the same as Unsigned64. Its interpretation is based on IETF RFC 5905 and expresses the network time related to a particular date as shown in Figure 2.

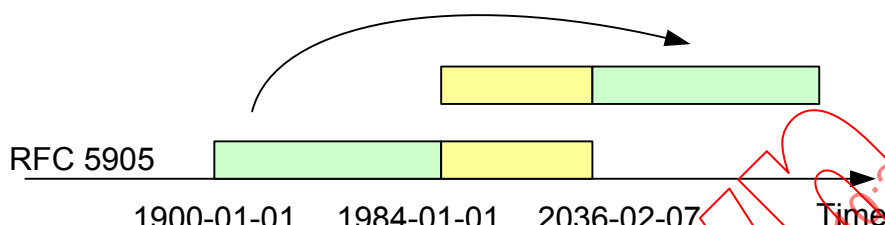


Figure 2 – NetworkTime date relation

The value of NetworkTime describes a time based on the UTC time zone (Universal Time Coordinated, also called Greenwich Mean Time) in units of $1/2^{32}$ seconds since 1900-01-01 T 00:00:00 Z for time values greater or equal to 0x9DFF440000000000 or else since 2036-02-07 T 06:28:16 Z for time values less than 0x9DFF440000000000, which represents the 1984-01-01 T 00:00:00 Z.

NOTE The time is represented according to ISO/IEC 8601.

Rollovers after 136 years are not automatically detectable and are to be maintained by the application.

The least significant Bit of the fractional portion (2^0) is used according to Table 22.

Table 22 – Status least significant Bit of the fractional portion (2^0)

Value	Meaning
0	NetworkTime related to the global synchronized time
1	NetworkTime related to the local time

5.3.1.7.3 NetworkTimeDifference

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 59
- 2 Data type Name = NetworkTimeDifference
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 8

This data type is the same as Integer64, i.e. a two's complement binary number with a length of eight octets. The value is interpreted as the difference between two NetworkTime values as fractional portions of seconds in $1/2^{32}$ s.

5.3.1.7.4 TimeStamp

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 60

2	Data type Name	=	TimeStamp
3	Format	=	STRUCTURE
3.1	Number of Fields	=	4
3.2.1	Field Name	=	Status
3.2.2	Field Data type	=	Unsigned16
3.2.3	Field Name	=	SecondsHigh
3.2.4	Field Data type	=	Unsigned16
3.2.5	Field Name	=	SecondsLow
3.2.6	Field Data type	=	Unsigned32
3.2.7	Field Name	=	Nanoseconds
3.2.8	Field Data type	=	Unsigned32

This data type defines a 12 octet fixed length data type.

The value is interpreted according to the following formula:

$$\text{TimeStamp.Seconds} = (\text{SecondsHigh} \ll 32) + \text{SecondsLow} \quad (14)$$

$$\text{TimeStamp.Nanoseconds} = \text{Nanoseconds}$$

where

<i>TimeStamp.Seconds</i>	is the seconds part of the calculated timestamp value (a 48 bit value)
<i>TimeStamp.Nanoseconds</i>	is the nanoseconds part of the calculated timestamp value
<i>SecondsHigh</i>	is the raw value for interpretation
<i>SecondsLow</i>	is the raw value for interpretation
<i>Nanoseconds</i>	is the raw value for interpretation

A TimeStamp is build according Table 23 and the following definitions:

- If the device is synchronized and in sync with the global time master the TimeStamp is related to the global time,
- Otherwise if the device has means of a local time, the TimeStamp is related to the local time.

Table 23 – Status value range

Value	Meaning
0	TimeStamp related to the global synchronized time
1	TimeStamp related to the local time
2	TimeStamp related to the local time (arbitrary time scale)
other	reserved

If the time is not battery buffered while the device is in power off, the status is set to “local time arbitrary time scale” until the device is synchronized to the global time.

If the synchronzation for the global time is lost, the status changes to “local time” until power off.

5.3.1.8 VisibleString character types

NOTE intentionally blank

5.3.2 String types

5.3.2.1 OctetString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	10
2	Data type Name	=	OctetString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

An OctetString is an ordered sequence of octets, numbered from 1 to n. For the purposes of discussion, octet 1 of the sequence is referred to as the first octet. IEC 61158-6-10 define the order of transmission.

NOTE IEC 61158-6-3 follows the definition of IEC 61158-6-10.

5.3.2.2 VisibleString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	9
2	Data type Name	=	VisibleString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

This type is defined as the ISO/IEC 646 International Reference Version string type without the "del" (coding 0x7F) character.

5.3.2.3 UNICODEString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	39
2	Data type Name	=	UNICODEString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

This type is defined as the ISO/IEC 10646 string type with all coding variants e.g. UTF-8, UTF-16 and UTF-32.

5.3.3 Structure types

5.3.3.1 OctetString2+Unsigned8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	103
2	Data type Name	=	OctetString2+Unsigned8
3	Format	=	STRUCTURE
5.1	Number of Fields	=	2
5.2	List of Fields		
5.2.1	Field Name	=	Value
5.2.2	Field Data type	=	OctetString[2]
5.2.3	Field Name	=	Status
5.2.4	Field Data type	=	Unsigned8

This data type defines the OctetString2+Unsigned8 and is coded according to Table 24.

Table 24 – OctetString2+Unsigned8

Bits	7	6	5	4	3	2	1	0
Octet 1	1. octet							
Octet 2	2. octet							
Octet 3	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

This data type is structured as follows.

Value

This field contains the value as OctetString with the length of 2.

Status

This field describes the status of the value as a qualifier.

5.3.3.2 Float32+Unsigned8

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 101
- 2 Data type Name = Float32+Unsigned8
- 3 Format = STRUCTURE
- 5.1 Number of Fields = 2
- 5.2 List of Fields
 - 5.2.1 Field Name = Value
 - 5.2.2 Field Data type = Float32
 - 5.2.3 Field Name = Status
 - 5.2.4 Field Data type = Unsigned8

This data type defines the Float32+Unsigned8 and is coded according to Table 25.

SN: sign 0 = positive, 1 = negative

Table 25 – Float32+Unsigned8 octets

Bits	7	6	5	4	3	2	1	0
Octet 1	Exponent (E)							
	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Octet 2	(E) Fraction (F)							
	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Octet 3	Fraction (F)							
	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
Octet 4	Fraction (F)							
	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}
Octet 5	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

This data type is structured as follows.

Value

This field contains the value as Float32.

Status

This field describes the status of the value as a qualifier.

5.3.3.3 Unsigned8+Unsigned8

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 102
- 2 Data type Name = Unsigned8+Unsigned8
- 3 Format = STRUCTURE
- 5.1 Number of Fields = 2
- 5.2 List of Fields
- 5.2.1 Field Name = Value
- 5.2.2 Field Data type = Unsigned8
- 5.2.3 Field Name = Status
- 5.2.4 Field Data type = Unsigned8

This data type defines the Unsigned8+Unsigned8 and is coded according to Table 26.

Table 26 – Unsigned8+Unsigned8 octets

Bits	7	6	5	4	3	2	1	0
Octet 1	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Octet 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

This data type is structured as follows.

Value

This field contains the value as Unsigned8.

Status

This field describes the status of the value as a qualifier.

5.3.3.4 Unsigned16_S

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 104
- 2 Data type Name = Unsigned16_S
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

This data type is structured and coded according to Table 27 and Table 28.

Table 27 – Unsigned16_S octets

Bits	7	6	5	4	3	2	1	0
Octets 1	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6
Octets 2	2^5	2^4	2^3	2^2	2^1	2^0	St1	St0

Table 28 – Unsigned16_S meaning

St1 (bit1)	St0 (bit0)	Meaning
0	0	input channel: bad (value is fail-safe value) output channel: reserved
0	1	input channel: simulation output channel: reserved

St1 (bit1)	St0 (bit0)	Meaning
1	0	input channel: uncertain output channel: reserved
1	1	input channel: good output channel: reserved

5.3.3.5 Integer16_S

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 105
- 2 Data type Name = Integer16_S
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

This data type is structured and coded according to Table 29 and Table 30.

SN: sign 0 = positive, 1 = negative

Table 29 – Integer16_S octets

Bits	7	6	5	4	3	2	1	0
Octets 1	SN	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6
Octets 2	2^5	2^4	2^3	2^2	2^1	2^0	St1	St0

Table 30 – Integer16_S meaning

St1 (bit1)	St0 (bit0)	Meaning
0	0	input channel: bad (value is fail-safe value) output channel: reserved
0	1	input channel: simulation output channel: reserved
1	0	input channel: uncertain output channel: reserved
1	1	input channel: good output channel: reserved

5.3.3.6 Unsigned8_S

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 106
- 2 Data type Name = Unsigned8_S
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 1

This data type is structured and coded according to Table 31 and Table 32.

Table 31 – Unsigned8_S octets

Bits	7	6	5	4	3	2	1	0
Octets 1	2^5	2^4	2^3	2^2	2^1	2^0	St1	St0

Table 32 – Unsigned8_S meaning

St1 (bit1)	St0 (bit0)	meaning
0	0	input channel: bad (value is fail-safe value) output channel: reserved
0	1	input channel: simulation output channel: reserved
1	0	input channel: uncertain output channel: reserved
1	1	input channel: good output channel: reserved

5.3.3.7 OctetString_S**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 107
 2 Data type Name = OctetString_S
 3 Format = STRING
 4.1 Octet Length = N

This data type is structured and coded according to Table 33 and Table 34.

Table 33 – OctetString_S octets

Bits	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Octet 1	ch(8)	ch(7)	ch(6)	ch(5)	ch(4)	ch(3)	ch(2)	ch(1)
***	***	***	***	***	***	***	***	***
Octet m	ch(n)	ch(n-1)	ch(n-2)	ch(n-3)	ch(n-4)	ch(n-5)	ch(n-6)	ch(n-7)
Octet m+1	st1(4)	st0(4)	st1(3)	st0(3)	st1(2)	st0(2)	st1(1)	st0(1)
***	***	***	***	***	***	***	***	***
Octet 3*m	st1(n)	st0(n)	st1(n-1)	st0(n-1)	st1(n-2)	st0(n-2)	st1(n-3)	st0(n-3)

ch(x) value for channel x; st(x) status information for channel x; 1<x≤n

Table 34 – OctetString_S status bits

st1 (bit1)	st0 (bit0)	Meaning
0	0	input channel: bad (value is fail-safe value) output channel: reserved
0	1	input channel: simulation output channel: reserved
1	0	input channel: uncertain output channel: reserved
1	1	input channel: good output channel: reserved

5.3.3.8 F message trailer with 4 octets**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 110

- 2 Data type Name = F message trailer with 4 octets
 3 Format = FIXED LENGTH
 4.1 Octet Length = 4

This data structure consists of the status/control octet and 3 octets CRC parameters. This data type can be associated with input or output data up to 12 byte.

This data type is structured and coded according to Table 35.

Table 35 – F message trailer with 4 octets

Bits	7	6	5	4	3	2	1	0
Octet 1	Status/Control octet							
Octet 2	High-octet CRC							
Octet 3	CRC							
Octet 4	Low- octet CRC (least significant octet)							

5.3.3.9 F message trailer with 5 octets

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 111
 2 Data type Name = F message trailer with 5 octets
 3 Format = FIXED LENGTH
 4.1 Octet Length = 5

This data structure consists of the status/control octet and 4 octet CRC parameters. This data type can be associated with input or output data up to 122 octets.

This data type is structured and coded according to Table 36.

Table 36 – F message trailer with 5 octets

Bits	7	6	5	4	3	2	1	0
Octet 1	Status/Control octet							
Octet 2	1. octet CRC (most significant octet)							
Octet 3	2. octet CRC							
Octet 4	3. octet CRC							
Octet 5	4. octet CRC (least significant octet)							

5.4 Data type ASE service specification

There are no operational services defined for the type object.

6 Communication model for common services

6.1 Concepts

6.1.1 Overview

The concepts of IEC 61158-1 common elements are applied. This part of the specification defines common services, which are used for decentralized periphery. Furthermore, this part of the specification defines the usage of, refinements, or extensions to the referenced standards.

6.1.2 FAL ASEs

Figure 3 shows the common services FAL ASEs according to their OSI layer.

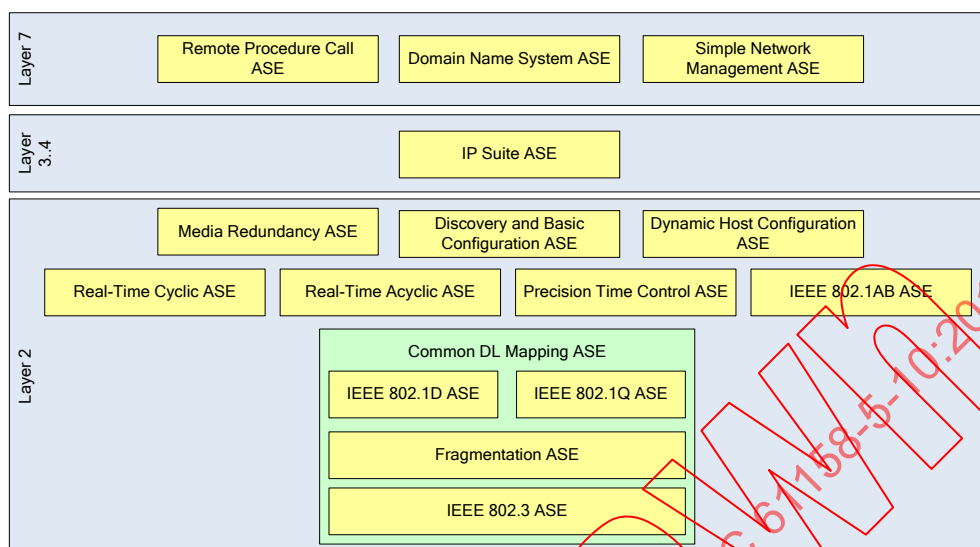


Figure 3 – FAL ASEs communication architecture

6.2 ASE data types

The data types supported by common services are a subset of those defined in Clause 5. They are:

- Boolean
- BinaryDate
- TimeOfDay with date indication
- TimeOfDay without date indication
- TimeDifference with date indication
- TimeDifference without date indication
- Float32
- Float64
- Integer8
- Integer16
- Integer32
- Integer64
- Unsigned8
- Unsigned16
- Unsigned32
- Unsigned64
- UUID
- NetworkTime
- NetworkTimeDifference
- OctetString
- VisibleString

6.3 Application Service Elements

6.3.1 Discovery and basic configuration ASE

6.3.1.1 Overview

The discovery and basic configuration is a concept of reading and writing basic network configuration parameters and discovering devices by means of different filter criteria. It uses IEEE 802.3 services.

The Discovery and basic Configuration Protocol (DCP) ASE is focused towards a particular Ethernet interface. Be aware that a particular device may have more than one Ethernet interface. The DCP ASE itself has no knowledge whether several interfaces belong to the same device; other ASE (e.g. simple network management ASE or else physical device management ASE) provide such means.

6.3.1.2 DCP class specification

6.3.1.2.1 General

The DCP ASE defines one DCP object type.

Some of the DCP ASE's attributes may be stored temporary or permanent as indicated by the service parameter Data Qualifier. If the Data Qualifier is set to "temporary" the attribute is reset to its default value after a power cycle. If the Data Qualifier is set to "permanent" the attribute contains its value during a power cycle. Thus the Data Qualifier determines the life time of the attributes value. If the attribute is set, its content is overwritten with the new value independent of the Data Qualifier. In particular there is exactly one attribute, which is overwritten independent of the Data Qualifiers value; e.g. a temporary setting overwrites thus the value of a previous permanent setting.

The DCP ASE contains elements which may be written by the Set service. Each of the writable elements has an additional attribute which has one of the following values:

YES

The write service to this element is always accepted. Thus the protocol layer handles the write service independent from the application.

NO

The write service to this element is always denied. Thus the protocol layer handles the write service independent from the application.

QUERY

The write service to this element needs the approval by the application. Thus the protocol and application layer handle the write service. In this case the local Set Command service is called by the protocol layer.

6.3.1.2.2 Template

A DCP object is described by the following template:

ASE:	DCP ASE
CLASS:	DCP
CLASS ID:	not used
PARENT CLASS:	IEEE 802.1AB
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: List of Device Options
2.1	(m) Attribute: Device Option
2.2	(m) Attribute: Device Suboption

3	(m) Attribute:	IP Option
3.1	(m) Attribute:	MAC Address
3.2	(m) Attribute:	IP Parameter
3.2.1	(m) Attribute:	IP Address
3.2.2	(m) Attribute:	Subnet Mask
3.2.3	(m) Attribute:	Standard Gateway
3.2.4	(m) Attribute:	DNS Server Addresses
4	(m) Attribute:	Device Properties Option
4.1	(m) Attribute:	Name Of Station
4.2	(m) Attribute:	List of Ports
4.2.1	(m) Attribute:	Alias Name Of Station
4.3	(m) Attribute:	Device Vendor
4.4	(m) Attribute:	Device ID
4.5	(m) Attribute:	Device Role
4.6	(m) Attribute:	Device Instance
5	(o) Attribute:	DHCP Option
5.1	(m) Attribute:	Host Name
5.2	(m) Attribute:	Vendor Specific Information
5.3	(m) Attribute:	Server Identifier
5.4	(m) Attribute:	Parameter Request List
5.5	(m) Attribute:	Class Identifier
5.6	(m) Attribute:	DHCP Client Identifier
5.7	(m) Attribute:	Fully Qualified Domain Name
5.8	(m) Attribute:	UUID/GUID-based Client Identifier
5.9	(m) Attribute:	Control DHCP for Address Resolution
6	(o) Attribute:	Device Initiative Option
6.1	(m) Attribute:	Device Initiative
7	(o) Attribute:	Manufacturer Specific Option
7.1	(m) Attribute:	List of Manufacturer Specific Suboptions
7.1.1	(m) Attribute:	Manufacturer Specific Suboption
7.1.2	(m) Attribute:	Manufacturer OUI
7.1.3	(m) Attribute:	Manufacturer Specific String
8	(m) Attribute:	Protocol machine Parameter
8.1	(m) Attribute:	Max Retry Limit
8.2	(m) Attribute:	UC Client Timeout
8.3	(m) Attribute:	MC Client Timeout
8.4	(m) Attribute:	TagControlInformation
8.4.1	(m) Attribute:	VLAN ID
8.4.2	(m) Attribute:	Priority
8.5	(m) Attribute:	Client Hold Time

SERVICES:

1	(m) OpsService:	Get
2	(m) OpsService:	Set
3	(m) OpsService:	Identify
4	(o) OpsService:	Hello

6.3.1.2.3 Attributes**Implicit**

The key attribute Implicit indicates that the DCP object is implicitly addressed by the service.

List of Device Options

This attribute list shall contain all options and suboptions which are supported by the device. It acts as an inventory over the supported DCP options and suboptions:

Device Option

This attribute contains an option that is supported by the device.

Attribute type: Unsigned8

Allowed values: IP, DEVICE_PROPERTIES, DHCP, CONTROL, DEVICE_INITIATIVE, MANUFACTURER_SPECIFIC_128, ... MANUFACTURER_SPECIFIC_254

Device Suboption

This attribute contains a suboption related to a specific option that is supported by the device.

Attribute type: Unsigned8

Allowed values:

On Device Option = IP:

MAC_ADDRESS, IP_PARAMETER, FULL_IP_SUITE

On Device Option = DEVICE_PROPERTIES:

DEVICE_VENDOR, NAME_OF_STATION, DEVICE_ID, DEVICE_ROLE,
DEVICE_OPTIONS, ALIAS_NAME, DEVICE_INSTANCE

On Device Option = DHCP:

DHCP_PARAMETER

On Device Option = CONTROL:

START, STOP, SIGNAL, FACTORY_RESET, RESET_TO_FACTORY

On Device Option = DEVICE_INITIATIVE:

DEVICE_INITIATIVE_STATE

On Device Option = MANUFACTURER_SPECIFIC:

MANUFACTURER_SPECIFIC_0, ... MANUFACTURER_SPECIFIC_255

IP Options

This attribute contains the following suboption attributes:

MAC Address

This attribute contains physical address of the Ethernet interface according to IEEE 802.3 MAC address.

The MAC Address shall be non volatile. It may be read by means of the Get service or it may be used within the Identify service as filter. It shall not be used as a suboption within the Set service.

Attribute type: OctetString[6]

IP Parameter

The IP Parameter shall have the same meaning as the corresponding attributes within the IP suite ASE as defined in 6.3.13. The service parameter Data Qualifier shall be used to address currently used or permanent stored attribute values for DCP services. This attribute list contains the following attributes:

IP Address

This attribute contains the non volatile IP address of the Ethernet interface according to IETF RFC 791 and IETF RFC 3330.

Attribute type: Unsigned32

Default Value: 0.0.0.0

Subnet Mask

This attribute contains the non volatile subnet mask of the Ethernet interface according to IETF RFC 791 and IETF RFC 3330.

Attribute type: Unsigned32

Default Value: 0.0.0.0

Standard Gateway

This attribute contains the non volatile IP address of the standard gateway according to IETF RFC 791 and IETF RFC 3330.

Attribute type: Unsigned32

Default Value: 0.0.0.0

DNS Server Addresses

This attribute contains up to 4 DNS server IP addresses. The value "0.0.0.0" indicates no address.

Attribute type: Unsigned32[4]

Default Value: 0.0.0.0

Device Properties Option

This attribute contains the following suboption attributes:

Name Of Station

This attribute contains the name of the interface provided by engineering. The value of this attribute shall be the same as for the attribute Chassis ID of the IEEE 802.1AB ASE. The values "port-xyz" or "port-xyz-rstuv" where x,y,z is in the range "0"-"9" from 001 up to 255 and r, s, t, u, v is in the range "0"-"9" from 00000 up to 65535 shall not be used.

NOTE 1 Be aware that the Name Of Station is in contrast to its misleading name a Name Of Interface, since as stated above the focus of the DCP ASE is a particular Ethernet interface.

Attribute type: VisibleString[240]

List of Ports

This attribute contains for each of the interface's ports the following attributes.

Alias Name Of Station

This attribute contains the alias names of the interface derived from Link Layer Discovery Protocol (LLDP) options. A particular alias shall concatenate the Peer Port ID, " and Peer Chassis ID. For each of the ports up to one alias name exists. A NULL string (i.e. "") indicates that no alias name exists on this port.

The Alias Name Of Station may only be used as filter in the Identify service. The services Get, Set and Hello do not offer access to this attribute.

NOTE 2 Example for Alias Name Of Station = "port-001.mill-1.factory3.org".

Attribute type: VisibleString[255]

Device Role

This attribute contains a bit map to indicate the roles of the device on this Ethernet interface.

The device roles indicate the possible roles of the device in the context of PROFINET IO. IO_DEVICE indicates that the Ethernet interface may provide up to one IO device instance, IO_MULTIDEVICE that it may provide even more than one IO device instance, the role IO_CONTROLLER that it may act as IO controller, and the role IO_SUPERVISOR that it may act as IO supervisor.

Attribute type: Unsigned8

Allowed values: IO_DEVICE, IO_CONTROLLER, IO_MULTIDEVICE, IO_SUPERVISOR

Device Vendor

This attribute contains a vendor specific string. It may be the type of the device or an ordering number.

Attribute type: VisibleString

Device ID

This attribute contains the Device Ident Number as described in 7.6.1.

Attribute type: Unsigned32

Device Instance

This attribute contains the instance or node number of the IO device's object UUID.

Attribute type: Unsigned16

DHCP Option

This optional attribute contains the following suboption attributes:

Host Name

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

Vendor Specific Information

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

Server Identifier

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

Parameter Request List

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

Class Identifier

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

DHCP Client Identifier

This attribute contains the client identifier for the usage of Dynamic Host Configuration Protocol (DHCP)

Attribute type: see IETF RFC 2132

Fully Qualified Domain Name

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

UUID/GUID-based Client Identifier

This attribute contains the value according to IETF RFC 2132.

Attribute type: see IETF RFC 2132

Control DHCP for Address Resolution

This attribute contains control information for the usage of Dynamic Host Configuration Protocol (DHCP).

Attribute type: Unsigned8

Allowed Values: DONT_USE_DHCP, DONT_USE_AND_RESET_DHCP_OPTIONS, USE_DHCP

Device Initiative Option

This optional attribute contains the following suboption attributes:

Device Initiative

This attribute contains the value to enable or disable the issuing of the Hello service.

Attribute type: Unsigned16

Allowed values: ON, OFF

Manufacturer Specific Option

This optional attribute contains the following suboption attributes:

List of Manufacturer Specific Suboptions

This attribute list contains the following attributes:

Manufacturer Specific Suboption

This attribute contains the manufacturer specific value.

Attribute type: Unsigned8

Manufacturer OUI

This attribute contains the organizational unique identifier to identify the vendor.

Attribute type: OctetString[3]

Manufacturer Specific String

This attribute contains the individual information.

Attribute type: OctetString

Protocol machine Parameter

This attribute contains the following attributes to describe the behavior of the protocol machines:

Max Retry Limit

This attribute contains the maximum limit to repeat a client service in case no answer has been received by the client.

Attribute type: Unsigned8

Default Value: 4

Allowed Values: 0 to 15

UC Client Timeout

This attribute contains the maximum value in seconds, which a client waits for a unicast response.

Attribute type: Unsigned16

Default Value: 1

Allowed Values: 1 to 30

MC Client Timeout

This attribute contains the maximum value in milliseconds, which a client waits for all possible identify responses. It shall be greater than the calculated Response Delay Time using the Response Delay Factor from the request. It shall be 400 ms if the Response Delay Factor equals one and rounded up to a full second plus 1 s otherwise.

Attribute type: Unsigned16

Allowed Values: 400 to 65 000

Tag Control Information

This attribute contains the following attributes:

VLAN ID

This attribute contains the VLAN ID according to IEEE 802.1Q.

Attribute type: Unsigned16

Allowed Value: 0

Priority

This attribute contains the priority tag of the frame according to IEEE 802.1Q.

Attribute type: Unsigned8

Allowed Value: 0

Client Hold Time

This attribute contains the maximum value in seconds, which a server shall only allow Get or Set services from the last active client. Service requests from other server shall be ignored before.

Attribute type: Unsigned16

Default Value: 3

6.3.1.3 DCP service specification

6.3.1.3.1 Get

The Get service is used to read one or more ASE attributes by means of the client server communication model. The attribute group shall be addressed by means of an option. The subgroup shall be addressed by means of a suboption.

The server shall respond with the individual status “not supported” if optional options or suboptions are not available. Generally, the requested data shall be responded as long as they fit in the DCP-Get-ResPDU. Otherwise, the status “resource error” for the first not fitting suboption shall be responded. If the requested option with suboption contains a list with more than one element then all available list elements shall be responded as long as they fit in the Protocol Data Unit (PDU). The element order is arbitrary. Table 37 shows the parameters of the service.

Table 37 – Get

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
DA	M	M(=)		
List Of Options	M	M(=)		
Option	M	M(=)		

Parameter name	req	ind	rsp	cnf
Suboption	M	M(=)		
Result(+)			S	S(=)
List Of Data			M	M(=)
Option			M	M(=)
Suboption			M	M(=)
Length			M	M(=)
Status			M	M(=)
IP Info			C	C(=)
Data			C	C(=)
AddUserData			C	C(=)
Result(-)			S	S(=)
ERRCLS			M	M(=)
ERRCODE			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

DA

This parameter shall contain the individual IEEE 802.3 MAC address of the server.

List Of Options

This parameter contains the requested list with options with their related suboptions. These parameters shall correspond with the ASE attributes.

Option

This parameter contains the option that shall be read from the device. It can contain the following ASE attributes: IP, Device Properties, DHCP, and Manufacturer Specific. The Control option shall not be used within this service. Furthermore, the All Selector option can be requested as a proxy option to get all available information from the server. In this case, the response shall at least contain the option IP with the suboption IP Parameter and the option Device Parameter with the suboptions Device ID, Device Role, Device Options, and Name Of Station. The response may contain further options/suboptions.

Suboption

This parameter contains the suboption corresponding with the option that shall be read from the device. Depending on the option used, only designated suboptions, which related to the grouped ASE attributes, shall be used.

Result(+)

This parameter indicates that the service request succeeded. It shall contain the requested values or an error code if an option/suboption is not available.

List Of Data

This parameter shall contain all values of the requested options with suboptions.

Option

This parameter contains the option requested from the device.

Suboption

This parameter contains the suboption requested from the device.

Length

This parameter contains the number of octets of the parameter Data and optional AddUserData depending of the parameter Status of the service response.

Status

This parameter contains the status of the parameter Data.

Allowed values: NO_ERROR – requested data fully delivered,
OPTION_NOT_SUPPORTED – requested optional Option is not supported,
SUBOPTION_NOT_SUPPORTED – requested optional suboption is not supported,
RESSOURCE_ERROR – requested available data does not fully fit in the DCP-Get-ResPDU

The status RESSOURCE_ERROR indicates that the value of an option could not be delivered completely in case it exceeds the available size of the PDU. However, the data should be present as long as it makes sense from the application point of view. For example, if the elements of the parameter List of Alias Names as a whole exceed the size of the PDU then, however, those elements which already fit should be present to deliver the requested information at least partly.

IP Info

This optional parameter contains the additional info in case of suboption IP Parameter and the parameter Status contains the value NO_ERROR.

Allowed values: IP_VIA_SET – IP Parameter via Set service received,
IP_VIA_DHCP – IP Parameter via DHCP service received,
IP_NOT_ACTIVE – IP Parameter not active, address conflict detected

Data

This parameter contains the value of the requested ASE attribute as long as it fits completely in the DCP-Get-ResPDU. This parameter shall only be present if the Status contains the value NO_ERROR.

AddUserData

This optional parameter can contain vendor specific data in a vendor specific format. This optional response parameter may only be present if the Status contains a value other than NO_ERROR.

Result(-)

This parameter indicates that the service request failed in general for local reasons.

ERRCLS

The parameter ERRCLS contains the error class of the specific error.

Type: Unsigned16

Allowed Values: CTXT – local context invalid, PROTOCOL – transmission error

ERRCODE

The parameter ERRCODE contains the error code of the specific error.

Type: Unsigned16

Allowed Values: INVALID_STATE if ERRCLS = CTX, LMPM if ERRCLS = PROTOCOL (local error during transmission), TIMEOUT if ERRCLS = PROTOCOL (remote station was not responding)

6.3.1.3.2 Set

The Set service is used to write one or more ASE attributes by means of the client server communication model. The attribute group shall be addressed by means of an option. The subgroup shall be addressed by means of a suboption.

Furthermore, special control commands control the behavior of the server. The commands Start Transaction and Stop Transaction shall be used to group a sequence of set service and the server shall be locked for a client in this case. The command Signal allows with its suboption FlashOnce to identify the address hardware visually. The command Reset To Factory Settings allows with its suboptions to reset in particular the application data, communication parameters, engineering parameters, all parameters, the complete device, or the firmware.

The server shall respond with the individual status “not supported” if optional options or suboptions are not available. The element order is arbitrary. Table 38 shows the parameters of the service.

Table 38 – Set

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
DA	M	M(=)		
List Of Data	U	U(=)		
Option	M	M(=)		
Suboption	M	M(=)		
Length	M	M(=)		
Data Qualifier	M	M(=)		
Manufacturer OUI	U	U(=)		
Data	M	M(=)		
List Of Control Commands	U	U(=)		
Control	M	M(=)		
Start Transaction	S	S(=)		
Stop Transaction	S	S(=)		
Factory Reset Settings	U	U(=)		
Reset To Factory Settings	U	U(=)		
Signal	U	U(=)		
FlashOnce	M	M(=)		
Result(+)			S	S(=)
List Of Response			M	M(=)
Option			M	M(=)
Suboption			M	M(=)
Length			M	M(=)
Status			M	M(=)
Add User Data			C	C(=)
Result(-)			S	S
ERRCLS			M	M
ERRCODE			M	M

Argument

The argument shall convey the service specific parameters of the service request.

DA

This parameter shall contain the individual IEEE 802.3 MAC address of the server.

List Of Data

This parameter contains the data to be written with options with their related suboptions. These parameters shall correspond with the ASE attributes.

Option

This parameter contains the option that shall be written to the server. It can contain the following ASE attributes: IP, Device Properties, DHCP, and Manufacturer Specific. The option IP with the suboption MAC Address shall not be used.

Suboption

This parameter contains the suboption corresponding with the option that shall be written to the server. Depending on the option used, only designated suboptions, which related to the grouped ASE attributes, shall be used.

Length

This parameter contains the number of octets of the parameter.

Data Qualifier

This parameter contains an additional info of the data that has to be written to the server. Depending on the requested option/suboption the status has different meanings.

Allowed Values for Option IP: USE_TEMPORARY AND CLEAR_STORED_ADDRESS, SAVE_PERMANENT

Allowed Values for other Option: USE_TEMPORARY, SAVE_PERMANENT

Manufacturer OUI

This optional parameter contains the OUI for the manufacturer specific option.

Data

This parameter contains the data that has to be written to the server.

List Of Control Commands

This parameter contains the following parameter:

Control

This parameter contains the following suboption parameter:

Start Transaction

This parameter shall mark the start of a sequence of different Set services. The server shall lock the ASE attributes for the requesting client until the Stop Transaction is indicated or a certain timeout occurs.

Attribute type: Unsigned8

Allowed Value: START

Stop Transaction

This parameter shall mark the end of a sequence of different Set services. The server shall unlock the ASE attributes.

Attribute type: Unsigned8

Allowed Value: STOP

Factory Reset Settings

This parameter shall be used to reset all DCP attributes by means of the Set services. The behavior onto the device's application is not specified. Out of these reasons this parameter is obsolete and substituted by the Reset To Factory Settings below.

Attribute type: Unsigned8

Allowed Value: FACTORY_RESET

Reset To Factory Settings

This parameter shall be used to reset to device according to the specified qualifier. Note that legal issues and application specific issues are defined by profiles and the local application.

Attribute type: Unsigned8

Allowed Value: RESET_FACTORY_SETTINGS

Reset application data

Reset data which has been stored permanent in submodules and modules to factory values.

- Manufacturer specific Record-Data shall be set to factory values.
- All I&M1 – I&M15 shall be set to factory values.

The support of this suboption is recommended.

Attribute type: Unsigned16

Allowed Value: RESET_APPLICATION_DATA

Reset communication parameters

All parameters active for the interface or the ports and the ARs shall be set to the default values and if permanently stored reset.

This mode is intended to set the addressed communication interface of a device into a state which almost similar to the "out of the box" state.

- NameOfStation shall be set to "".
- IP suite parameter shall be set to 0.0.0.0
- DHCP parameters (if available) shall be reset
- All PDev parameters (PD IR Data, PD Port Data Adjust, PD Interface MRP Data Adjust...) shall be set to factory values.

The support of this suboption is mandatory.

Attribute type: Unsigned16

Allowed Value: RESET_COMMUNICATION_PARAMETER

Reset engineering parameters

Reset engineering parameters which have been stored permanent in the IOC or IOD to factory values.

- If a node is able to switch its functionality from IOC to IOD and vice versa via engineering system, the factory functionality shall be activated.
- An application program loaded by an engineering system shall be reset (or removed).
- A configuration loaded by an engineering system shall be reset (or removed).

The support of this suboption is recommended.

If an IOC or IOD has not such kind of parameters it shall respond with success.

Attribute type: Unsigned16

Allowed Value: RESET_ENGINEERING_PARAMETER

Reset all parameters

Resets all stored data in the IOD or IOC to its factory values.

- If a node is able to switch its functionality from IOC to IOD and vice versa via engineering system, the factory functionality shall be activated.
- An application program loaded by an engineering system shall be reset (or removed).
- A configuration loaded by an engineering system shall be reset (or removed).

The support of this suboption is recommended.

Attribute type: Unsigned16

Allowed Value: RESET_ALL_PARAMETER

NOTE 1 This covers all data related to RESET_APPLICATION_DATA, RESET_COMMUNICATION_PARAMETER and RESET_ENGINEERING_PARAMETER.

Reset device

Resets all stored data in the IOD or IOC to its factory values.

This service shall reset the communication parameters of all interfaces of the device and should reset all parameters of the device.

This mode is intended to set the device into a state which is similar to the “out of the box” state.

The support of this suboption is mandatory.

Attribute type: Unsigned16

Allowed Value: RESET_DEVICE

NOTE 2 Same behavior as RESET_COMMUNICATION_PARAMETER but covers also other interfaces and additionally (if supported) as RESET_ALL_PARAMETERS.

Reset firmware

Reset installed software revisions to factory images.

The support of this suboption is optional.

Attribute type: Unsigned16

Allowed Value: RESET_FIRMWARE

Signal

This parameter contains the following parameter:

Flash Once

This parameter shall contain the signal value. Flash Once means to flash the Ethernet LINK LED (or an alternative signalization) with a duration of 3 s with a frequency of 1 Hz (500 ms on, 500 ms off).

Attribute type: Unsigned16

Allowed Value: FLASH_ONCE

Result(+)

This parameter indicates that the service request succeeded. It shall contain the requested values.

List Of Response

This parameter contains the return values of the write operation of the requested option/suboption.

Option

This parameter contains the option requested from the device.

Suboption

This parameter contains the suboption requested from the device.

Length

This parameter contains the number of octets of the parameter Data and optional AddUserData depending of the parameter Status of the service response.

Status

This parameter contains the status of the parameter Data transport. The value NO_ERROR signals the delivery and if requested the permanent storage of the data. It may take some time before the parameter is working.

Allowed values: NO_ERROR – requested data fully delivered,
OPTION_NOT_SUPPORTED – requested optional Option is not supported,
SUBOPTION_NOT_SUPPORTED – requested optional suboption is not supported,
SET_NOT_POSSIBLE – set not possible for local reasons,
IN_OPERATION – set not possible because of operating application

AddUserData

This optional parameter can contain vendor specific data in a vendor specific format. This optional response parameter may only be present if the Status contains a value other than NO_ERROR.

Result(-)

This parameter indicates that the service request failed in general for local reasons.

ERRCLS

The parameter ERRCLS contains the error class of the specific error.

Type: Unsigned16

Allowed Values: CTXT – local context invalid, PROTOCOL – transmission error

ERRCODE

The parameter ERRCODE contains the error code of the specific error.

Type: Unsigned16

Allowed Values: INVALID_STATE if ERRCLS = CTX, LMPM if ERRCLS = PROTOCOL (local error during transmission), TIMEOUT if ERRCLS = PROTOCOL (remote station was not responding)

6.3.1.3.3 Local Set Command

The local Set Command service is used to indicate a Set service to the FAL, if the attribute of the element is set to **QUERY** (see 6.3.1.2.1).

The service passes the arguments of the Set service; thus for further description regarding the arguments see 6.3.1.3.2.

Table 39 shows the parameters of the service.

Table 39 – Local Set Command

Parameter name	ind	rsp
Argument	M	
DA	M	
List Of Data	U	

Parameter name	ind	rsp
Option	M	
Suboption	M	
Length	M	
Data Qualifier	M	
Manufacturer OUI	U	
Data	M	
List Of Control Commands	U	
Control	M	
Start Transaction	S	
Stop Transaction	S	
Factory Reset Settings	U	
Reset To Factory Settings	U	
Signal	U	
FlashOnce	M	
Result(+)		S
List Of Response		M
Option		M
Suboption		M
Length		M
Status		M
Add User Data		C
Result(-)		S
ERRCLS		M
ERRCODE		M

6.3.1.3.4 Identify

The Identify service can be used to discover devices on the network. The service shall be sent to a well-known multicast IEEE 802.3 MAC address that is defined by the DCP protocol. Only devices that fulfill all requested filter shall reply to this request. The Identify confirmation shall contain all responses received within a certain time interval. The responding devices are identified by the IEEE 802.3 source MAC address. Otherwise, the confirmation shall contain an empty list.

In case the filter criteria are matched a server shall respond with the data for options and suboptions that are part of the filter. In addition, the data of option

- IP with suboption IP Parameter and its elements IP Address, Subnet Mask, and Standard Gateway,
- Device Properties with its suboptions List of Station Names, Device ID, Device Role, and Device Option

shall always be included in the response. The sequence of data blocks may be arbitrary. Table 40 shows the parameters of the service.

Table 40 – Identify

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
List Of Filter	S	S(=)		
Option	M	M(=)		
Suboption	M	M(=)		
Length	M	M(=)		
Data	M	M(=)		
All Selector	S	S(=)		
Response Delay Factor	M	M(=)		
Result(+)			S	S(=)
List of Devices			C	C(=)
SA				M
List of Data			M	M(=)
Option			M	M(=)
Suboption			M	M(=)
Length			M	M(=)
Data			C	C(=)
Result(-)			S	S(=)
ERRCLS			M	M(=)
ERRCODE			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

List Of Filter

This conditional parameter contains the options/suboptions that shall be used to filter the request on the server side. More than one filter shall be handled with an AND operation at the server side.

NOTE AND operation means that a server only responds to an identify request within the response delay time if all filter criteria match with currently used attribute values.

Option

This parameter contains the value of the option that shall be used to filter the request on the server side. It can contain the following ASE attributes: IP, Device Properties, DHCP, and Manufacturer Specific.

Suboption

This parameter contains the value of the suboption that shall be used to filter the request on the server side. Depending on the option used, only designated suboptions, which related to the grouped ASE attributes, shall be used.

Length

This parameter contains the length of the data of the option and suboption that shall be used to filter the request on the server side.

Data

This parameter contains the data of the option and suboption that shall be used to filter the request on the server side.

All Selector

This conditional parameter indicates that all devices shall at least respond with their mandatory options.

Response Delay Factor

This parameter contains the factor for the server to calculate the spread value to delay the response if the List Of Filter criteria are fulfilled. The time base is 10 ms. The calculated delay of the server shall be between 10 ms and 64 s.

Parameter type: Unsigned16

Allowed Values: 1 to 6 400

Default Values: 1 (IO Controller)

Result(+)

This parameter indicates that the service request succeeded. It shall contain the requested values.

List of Devices

This parameter contains the list of devices returned from the identification.

SA

This parameter contains the individual IEEE 802.3 MAC address of the device.

List of Data

This parameter contains the data of the device specified by SA. It consists of the following elements:

Option

This parameter contains the option to identify the returned data.

Suboption

This parameter contains the suboption to identify the returned data.

Length

This parameter contains the length of the returned data.

Data

This parameter contains the data belonging to the specified option and suboption.

Result(-)

This parameter indicates that the service request failed in general for local reasons.

ERRCLS

The parameter ERRCLS contains the error class of the specific error.

Type: Unsigned16

Allowed Values: CTXT – local context invalid, PROTOCOL – transmission error

ERRCODE

The parameter ERRCODE contains the error code of the specific error.

Type: Unsigned16

Allowed Values: INVALID_STATE if ERRCLS = CTX, LMPM if ERRCLS = PROTOCOL (local error during transmission), TIMEOUT if ERRCLS = PROTOCOL (remote station was not responding)

6.3.1.3.5 Hello

The optional Hello service can be used to announce the presence of a device in a network. The service shall be sent to a well-known multicast IEEE 802.3 MAC address that is defined by the DCP protocol.

The service shall transmit the requested number of DCP-Hello-ReqPDUs within a certain restricted time period. The requested hello interval shall be used to wait before each retransmission.

The Hello service is intended to use to accelerate the startup phase therefore transmission should not remain longer than a view hundred milliseconds.

Table 41 shows the parameters of the service.

Table 41 – Hello

Parameter name	req	ind	cnf
Argument	M	M(=)	
List of Data	M	M(=)	
Name of Station	M	M(=)	
IP Parameter	M	M(=)	
IP Address	M	M(=)	
Subnet Mask	M	M(=)	
Standard Gateway	M	M(=)	
Device ID	M	M(=)	
List of Device Options	M	M(=)	
Device Option	M	M(=)	
Device Suboption	M	M(=)	
Length	M	M(=)	
Data	M	M(=)	
Device Role	M	M(=)	
Number of DCP-Hello-ReqPDUs	M		
Initial Delay	M		
Hello Interval	M		
Result(+)			S(=)
Result(-)			S(=)

Argument

The argument shall convey the service specific parameters of the service request.

List Of Data

This parameter list contains of the following parameter.

Name Of Station

This parameter contains the value of the corresponding ASE attribute.

IP Parameter

This parameter consists of the following parameter:

IP Address

This parameter contains the value of the corresponding ASE attribute.

Subnet Mask

This parameter contains the value of the corresponding ASE attribute.

Standard Gateway

This parameter contains the value of the corresponding ASE attribute.

Device ID

This parameter contains the value of the corresponding ASE attribute.

List of Device Options

This parameter list consists of the following parameter:

Device Option

This parameter contains the value of the corresponding ASE attribute.

Device Suboption

This parameter contains the value of the corresponding ASE attribute.

Length

This parameter contains the length of the data.

Data

This parameter contains the data of the option and suboption.

Device Role

This parameter contains the value of the corresponding ASE attribute.

Number of DCP-Hello-ReqPDUs

This parameter contains the number of DCP-Hello-ReqPDUs that shall be issued with the service request.

Parameter type: Unsigned32

Allowed Values: 1 to 15

Default Values: 3

Initial Delay

This parameter contains the delay time for the first DCP-Hello-ReqPDU.

Time base: 1 ms

Parameter type: Unsigned32

Allowed Values: 0, 50, 100, 500, 1000

Hello Interval

This parameter contains the time interval between transmitting the first and each subsequent DCP-Hello-ReqPDU.

Parameter type: Unsigned16

Allowed Values: 30_MILLISECONDS, 50_MILLISECONDS, 100_MILLISECONDS, 300_MILLISECONDS, 500_MILLISECONDS, 1_SECONDS

Default Values: 30_MILLISECONDS

Result(+)

This parameter indicates that the service request succeeded. It shall contain the requested values.

Result(-)

This parameter indicates that the service request failed in general for local reasons.

6.3.2 Precision time control ASE**6.3.2.1 Concepts**

The "Precision Transparent Clock Protocol" (PTCP) represents a uniform method of distributing a time. In particular, this means that a PTCP master synchronizes an amount of PTCP slaves in the sub-microsecond-range. It also supports a PTCP best master model and up to 32 different PTCP domains. The transport protocol PTCP is data link layer based.

A PTCP domain shall be used for CLOCK, which is a local time to synchronize the network access (in particular the cyclic communication) within all devices participating in the particular PTCP domain.

No further PTCP domains are currently supported.

Additionally PTCP supports the measurement of the line delay between two corresponding ports. Figure 4 shows the context of the precision time control ASE.

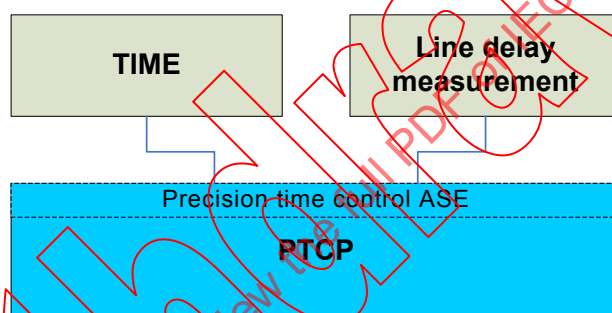


Figure 4 – PTCP applications

6.3.2.2 PTCP class specification**6.3.2.2.1 Template**

ASE: PTCP ASE

CLASS: PTCP

CLASS ID: not used

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Key Attribute:	Sync ID
2	(m) Attribute:	Domain
2.1	(m) Attribute:	Domain UUID
2.2	(m) Attribute:	Sequence ID
2.3	(m) Attribute:	Sync Send Factor
2.4	(m) Attribute:	PTCP Takeover Timeout
2.5	(m) Attribute:	PTCP Timeout
2.6	(m) Attribute:	Role (MASTER, SLAVE)
2.7	(c) Constraint:	Role = MASTER
2.7.1	(m) Attribute:	Priority
2.7.2	(m) Attribute:	Accuracy
2.7.3	(m) Attribute:	Variance
2.7.4	(m) Attribute:	PTCP Master Startup Time
2.8	(c) Constraint:	Sync ID = TIME

2.8.1	(m) Attribute:	PTCP Time
2.8.1.1	(m) Attribute:	Epoch Number
2.8.1.2	(m) Attribute:	Seconds
2.8.1.3	(m) Attribute:	Nano Seconds
2.8.1.4	(m) Attribute:	Current UTC Offset
2.9	(m) Attribute:	PLL Window
2.10	(m) Attribute:	Sync State Info
2.10.1	(m) Attribute:	Jitter out of Boundary
2.10.2	(m) Attribute:	No Sync Message Received
2.10.3	(m) Attribute:	Wrong PTCP Domain UUID
2.11	(m) Attribute:	Rate Compensation Factor
2.12	(m) Attribute:	List of Ports
2.12.1	(m) Attribute:	PortID
2.12.2	(m) Attribute:	Borderline
2.12.2.1	(m) Attribute:	Ingress
2.12.2.2	(m) Attribute:	Egress
2.12.3	(m) Attribute:	Line Delay
2.12.4	(m) Attribute:	Rate Compensation Factor Peer
2.13	(o) Attribute:	List of OUIs
2.13.1	(m) Attribute:	OUI
2.13.2	(s) Attribute:	Subtype
2.13.3	(s) Attribute:	Data Block

SERVICES:

1	(m) OpsService:	Start Bridge
2	(m) OpsService:	Start Slave
3	(o) OpsService:	Start Master
4	(m) OpsService:	Stop Bridge
5	(m) OpsService:	Stop Slave
6	(o) OpsService:	Stop Master
7	(m) OpsService:	Sync State Change

6.3.2.2.2 Attributes

Sync ID

This key attribute identifies the instance of the protocol machine to which the data belong. The value CLOCK shall be used for cycle synchronization.

Attribute type: Unsigned8

Allowed values: CLOCK

Domain

This attribute consists of the following elements:

Domain UUID

This attribute contains the UUID of the domain for synchronization which is provided by the engineering system and issued by a master. The UUID is used for a logical grouping of PTCP clocks that synchronize each other using the PTCP protocol. Clocks in another PTCP domains shall not be synchronized.

Attribute type: UUID

Sequence ID

This attribute contains the sequence number of the sync message. It shall be incremented with every new sync messages. By means of the Sequence ID the receiver detects duplications.

Attribute type: Unsigned16

Sync Send Factor

This attribute contains the send interval for sync messages.

Attribute type: Unsigned16

Allowed values: 0 to 64 000

PTCP Takeover Timeout

This attribute contains a timeout value for detection the loss of sync messages from the current sync master. The sync slave shall try to find a new sync master. The PTCP takeover timeout value shall be in the range from 32 ms to 16 352 ms.

Attribute type: Unsigned32

Default value: 96 ms

PTCP Timeout

This attribute contains a timeout value for detection the loss of any sync messages. The PTCP timeout value shall be in the range from 32 ms to 16 352 ms.

Attribute type: Unsigned32

Default value: 192 ms

Role

This attribute contains the role to time synchronization.

Attribute type: Unsigned16

Allowed values: MASTER, SLAVE

Priority

This attribute shall be the priority of master issuing sync messages.

Attribute type: Unsigned8

Allowed values: PRIMARY, SECONDARY

Accuracy

This attribute shall be the accuracy of master time in comparison with the precision of the local clock.

Attribute type: Unsigned8

Allowed values: 25_ns, 100_ns, 250_ns, 1_μs, 2,5_μs, 10_μs, 25_μs, 100_μs, 250_μs, 1_ms, UNKNOWN

Default value: 100_ns

Variance

This attribute shall be the value for the quality of the clock. Each master clock shall maintain an estimation of its inherit precision. The PTCP variance is based on the theory of Allan deviation. The Allan deviation is $\sigma_y(\tau)$ is estimated as follows:

$$\sigma_y(\tau) = \left[\frac{1}{2(N-2)\tau^2} \times \sum_{k=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \right]^{\frac{1}{2}} \quad (15)$$

where

$\sigma_y(\tau)$ is the Allan deviation.

x_k , x_{k+1} and x_{k+2} are time residual measurement, made at times t_k , $t_k + \tau$ and $t_k + 2\tau$, between the

time provided by the measured clock and a local reference clock.

N is the number of data samples.

The PTCP variance is defined by:

$$\sigma_{PTCP}^2 = \tau^2 \times \frac{1}{3} \sigma_y^2 \quad (16)$$

where

σ_{PTCP}^2 is PTCP variance.
 σ_y is the Allan deviation.
 τ is the quantity

An unbiased estimate of the PTCP variance shall be computed as follows:

$$\sigma_{PTCP}^2 = \frac{1}{3} \left[\frac{1}{2(N-2)} \times \sum_{k=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \right] \quad (17)$$

where

σ_{PTCP}^2 is the unbiased estimate of the PTCP variance
 x_k, x_{k+1} and x_{k+2} are time residual measurement, made at times $t_k, t_k + \tau$ and $t_k + 2\tau$, between the time provided by the measured clock and a local reference clock
 N is the number of data samples

For PTCP variance the quantity τ , the sample period, shall be the value defined as sync interval.

PTCP variances shall be represented as follows:

An estimate of the variance, σ_{PTCP}^2 , is computed in units of seconds squared.

The logarithm to the base 2 of this estimate is computed. The computation of the logarithm need not be more precise than the precision of the estimate of the variance.

The logarithm is multiplied by 28 to produce a scaled value.

The value is represented as a 2's complement Integer16. The value 0x8000 is added to the reported value represented in this form and any overflow is ignored. The result, i.e., the offset scaled reported value, is cast as an Unsigned16.

This offset scaled value, represented as Unsigned16, shall be the value of the log variances.

The largest possible positive number, 0xFFFF, for the offset scaled log variance attribute shall indicate that the variance is either too large to be represented, or has not been computed. The variance values are used in the selection of the best master clock.

Attribute type: Unsigned16

NOTE 1 For example, suppose the PTCP variance value is $1\,414 \times 2^{-73} = 1\,497 \times 10^{-22} \text{ s}^2$. Then, $\log_2(1\,414 \times 2^{-73}) = -73 + 0,5 = -72,5$. If this were expressed as an Integer16, it would truncate to -72. To retain some precision

the value is scaled by 2^8 to yield a scaledLogVariance of -18 560, 0xB780, which retains 8 bits more precision. To this is added 0x8000 to yield the offset scaled reported value 0x3780.

NOTE 2 The smallest variance that can be represented is 2^{-128} or $\sim 3 \times 10^{-39} \text{ s}^2$, which results in an offset scaled log variance of 0x0000. The maximum variance that can be represented is $\sim 2^{+127.99609}$, which results in an offset scaled log variance of 0xFFFF.

PTCP Master Startup Time

This attribute shall be the duration of the startup phase of the sync master. During this time the sync master shall check if there is a sync master with a higher clock quality in the sync domain.

Attribute type: Unsigned16

Allowed values: 0 to 300 s

Default value: 10 s

Time

This attribute consists of the following elements for the time representation:

PTCP Time

This attribute consists of the following elements for the PTCP time representation since the begin of the PTCP epoch 1970-01-01T00:00:00Z.

Epoch Number

This attribute shall be the current number of times the 32-bit Seconds clock has rolled over since the begin of the PTCP epoch.

Attribute type: Unsigned16

Seconds

This attribute contains the current number of seconds since 1970-01-01T00:00:00Z.

Attribute type: Unsigned32

Nano Seconds

This attribute contains the current number of nanoseconds until the next second.

Attribute type: Unsigned32

Allowed values: 0 to 999 999 999

Current UTC Offset

This attribute contains the current number of leap seconds as offset from the PTCP Time to the Coordinated Universal Time (UTC).

Attribute type: Integer16

Clock

This attribute consists of the following elements for the clock representation:

Local Time

This attribute consists of a local time value in nanoseconds. This value shall be used for PTCP clock synchronization and shall be the basis for the cycle counter.

Attribute type: Unsigned64

PLL Window

This attribute shall characterize the synchronization accuracy. The deviation between the local time and the reference time issued by a master shall not exceed the value of the PLL Window.

Attribute type: Unsigned32

Sync State Info

This element consists of the following error attributes. The synchronization achieved the state synchronous if there is no error.

Jitter out of Boundary

This attribute identifies whether the necessary synchronization accuracy (inside or outside PLL Window) is achieved or not.

Attribute type: Boolean

No Sync Message Received

This attribute identifies whether synchronization messages have been received or not.

Attribute type: Boolean

Wrong PTCP Domain UUID

If an synchronization messages with wrong Domain UUID have been received, this attribute will be set.

Attribute type: Boolean

Rate Compensation Factor

This attribute shall be used to adjust the bridging time from sync and delay messages.

Attribute type: Unsigned32

List of Ports

This attribute consists of the following elements:

PortID

This attribute identifies one Port of a bridge.

Attribute type: Unsigned8

Borderline

Both subsequent attributes shall contain the value ON for end-ports of a PTCP Domain. The value of the Ingress attribute shall be ON and the value for the Egress attribute shall be OFF for an port coupling two PTCP segments, for the hierarchical higher segment. The value of the Ingress attribute shall be OFF and the value for the Egress attribute shall be ON for an port coupling two PTCP segments, for the hierarchical lower segment.

NOTE 3 It is intended to avoid transmission from PTCP sync and PTCP announce messages from a hierarchical lower segment to a higher segment within a PTCP domain

This attribute consists of the following elements:

Ingress

This attribute identifies the end-ports of a PTCP Domain.

Attribute type: Unsigned8

Allowed values: ON, OFF

Egress

This attribute identifies the end-ports of a PTCP Domain or a level / segment borderline.

Attribute type: Unsigned8

Allowed values: ON, OFF

Line Delay

This attribute contains the transmit latency from this port to the connected port.

Attribute type: Unsigned32

Rate Compensation Factor Peer

This attribute contains the ratio between local and remote crystal frequency of the device at the connected port.

Attribute type: Float64

List of OUIs

This attribute consists of the following elements:

OUI

This attribute is an Organizationally Unique Identifier as defined by IANA.

Attribute type: Octet[3]

Subtype

This attribute is an organizationally specific parameter valid in the context of the attribute OUI.

Attribute type: Unsigned8

Data Block

This attribute is an organizationally specific parameter valid in the context of the attributes OUI and Subtype.

Attribute type: UUID

6.3.2.3 PTCP service specification

6.3.2.3.1 Start bridge

The start bridge service creates an instance of the PTCP protocol machine. Table 42 shows the parameters of the service.

Table 42 – Start bridge

Parameter name	req	cnf
Argument	M	
Sync ID	M	
List of Port Parameter	U	
Port ID	M	
Borderline	M	
Result(+)		S
Sync ID		M
Result(-)		S
Sync ID		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

List of Port Parameter

This optional list of parameter consists of the following parameter:

Port ID

This parameter contains the ID of the port which issues the delay request message

Borderline

This parameter contains the end-ports of a clock-domain

Result(+)

This parameter indicates that the service request succeeded.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Error code

The parameter Error code contains the error code of the specific error.

Type: Unsigned16

Allowed Values: ENTRY_NOT_POSSIBLE, SYNC_ID_EXISTS

6.3.2.3.2 Start slave

The start slave service adds synchronization properties to the PTCP protocol machine with. Table 43 shows the parameters of the service.

Table 43 – Start slave

Parameter name	req	cnf
Argument	M	
SyncID	M	
Subdomain UUID	M	
Sync Send Factor	M	
PTCP Timeout Factor	M	
PLL Window	M	
Result(+)		S
Sync ID		M
Result(-)		S
Sync ID		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Subdomain UUID

This parameter contains the UUID of the synchronization subdomain. The UUID is used for a logical grouping of PTCP clocks that synchronize each other using the PTCP protocol.

Sync Send Factor

This parameter contains the reception interval of sync messages.

PTCP Timeout Factor

This parameter contains a timeout value to detect the loss of sync messages.

PLL Window

This parameter contains the required synchronization accuracy. The deviation between the local time and the reference time issued by a grandmaster shall not exceed the value of the PLLWindow.

Result(+)

This parameter indicates that the service request succeeded.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Error code

The parameter Error code contains the error code of the specific error.

Type: Unsigned16

Allowed Values: SLAVE_NOT_POSSIBLE, SLAVE_EXISTS

6.3.2.3.3 Start master

The start master service adds synchronization properties with the qualification to become synchronization master to the PTCP protocol machine. Table 44 shows the parameters of the service.

Table 44 – Start master

Parameter name	req	cnf
Argument	M	
Sync ID	M	
Subdomain UUID	M	
Sync Send Factor	M	
Sync Class	M	
Clock Stratum	M	
Current UTC Offset	M	
Epoch Number	M	
Result(+)		S
Sync ID		M
Result(-)		S
Sync ID		M

Parameter name	req	cnf
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Subdomain UUID

This parameter contains the UUID of the synchronization subdomain. The UUID is used for a logical grouping of PTCP clocks that synchronize each other using the PTCP protocol.

Sync Send Factor

This parameter contains the send interval of sync messages.

Sync Class

This parameter contains the recommended role (primary, secondary) as a master.

Clock Stratum

This parameter contains the quality of the own time.

Current UTC Offset

This parameter contains the current offset to the UTC.

Epoch Number

This parameter contains the current number of times the 32-bit Seconds clock has rolled over since the begin of the PTCP epoch. The PTCP epoch began at 0 hours on 1. January 1970.

Result(+)

This parameter indicates that the service request succeeded.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Error code

The parameter Error code contains the error code of the specific error.

Type: Unsigned16

Allowed Values: MASTER_NOT_POSSIBLE, MASTER_EXISTS, MASTER_NOT_EXISTS

6.3.2.3.4 Stop bridge

This service shall be used to stop the PTCP protocol machine. Table 45 shows the parameters of the service.

Table 45 – Stop bridge

Parameter name	req	cnf
Argument	M	
Sync ID	M	

Parameter name	req	cnf
Result(+)		S
Sync ID		M
Result(-)		S
Sync ID		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Error code

The parameter ErrCode contains the error code of the specific error.

Type: Unsigned16

Allowed Values: SYNC_ID_NOT_EXISTS

6.3.2.3.5 Stop slave

This service shall be used to stop the PTCP protocol machine. Synchronization will be stopped and bridge functionality remains. Table 46 shows the parameters of the service.

Table 46 – Stop slave

Parameter name	req	cnf
Argument	M	
Sync ID	M	
Result(+)		S
Sync ID		M
Result(-)		S
Sync ID		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

SyncID

This is the key attribute to identify the instance of the protocol machine.

Error code

The parameter Error code contains the error code of the specific error.

Type: Unsigned16

Allowed Values: SLAVE_NOT_EXISTS

6.3.2.3.6 Stop master

This service shall be used to reduce the properties of the PTCP protocol machine. Master functionality will be removed and the protocol machine is forced to takeover slave role. Table 47 shows the parameters of the service.

Table 47 – Stop master

Parameter name	req	cnf
Argument	M	
Sync ID	M	
Result(+)		S
Sync ID		M
Result(-)		S
Sync ID		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Result(-)

This parameter indicates that the service request failed.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Error code

The parameter Error code contains the error code of the specific error.

Type: Unsigned16

Allowed Values: MASTER_NOT_EXISTS

6.3.2.3.7 Sync state change

This service shall be used to indicate a change of the synchronization state (i.e. synchronized or out of synchronization) and/or a change in the synchronization quality (i.e. jitter out of boundary, no sync message received). A change can only appear for a synchronization slave. Table 48 shows the parameters of the service.

Table 48 – Sync state change

Parameter name	ind
Argument	M
Sync ID	M
Sync State	M
Reason List	C

Argument

The argument shall convey the service specific parameters of the service request.

Sync ID

This is the key attribute to identify the instance of the protocol machine.

Sync State

This is the attribute shall contain the value SYNCHRONIZED or NOT_SYNCHRONIZED.

Reason List

This conditional attribute consists of the following elements:

ErrorType

This attribute identifies a synchronization error.

Attribute type: Unsigned16

Allowed Values: JITTER_OUT_OF_BOUNDARY, NO_SYNC_MESSAGE_RECEIVED, WRONG_PTCP_DOMAIN

Appear

This attribute identifies whether the error appears or disappears.

Attribute type: Boolean

Allowed Values: TRUE, FALSE

6.3.2.3.8 Line Delay change

This local service shall be used to indicate a change of certain Local Systems Data. Table 49 shows the parameters of the service.

Table 49 – Line Delay change

Parameter name	ind
Argument	M
Local Port ID	M

Parameter name	ind
Local Line Delay	M

Argument

The argument shall convey the service specific parameters of the service indication.

Local Port ID

This parameter is the value of the corresponding attribute of the ASE object.

Local Line Delay

This parameter is the value of the corresponding attribute of the ASE object.

6.3.2.3.9 PTCP behavior

6.3.2.3.9.1 Synchronization

The synchronization is done with a sync- and an optional follow up-frame. The follow up-frame is necessary, if the hardware isn't able to add bridge and line delays on the fly.

The clock issuing the sync frames in a PTCP subdomain is called PTCP master.

Clocks which receive sync frames and adjust their time to the PTCP master are called PTCP slaves. Their synchronization accuracy depends on the accuracy of the time stamping units from each bridge which measures the line and bridge delays.

A bridge with the following enhancements is called a PTCP transparent clock:

- Line-Delay measurement
- Bridge Delay measurement
- Rate compensation for line and bridge delays
- Support media redundancy mechanism for PTCP

6.3.2.3.9.2 Rate control compensation

The rate control compensation equalizes the clock rate of a PTCP transparent clock to the PTCP master. Due to this the accumulated bridge and line delays, received by the PTSP slave to achieve higher synchronization accuracy, are measured with the clock rate of the PTCP master.

A PTCP transparent clock uses sync frames to measure the clock rate relating to the PTCP master. Figure 5 shows the clock drift measurement.

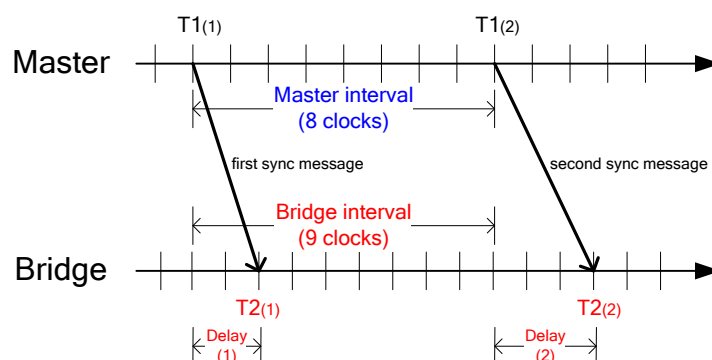


Figure 5 – Clock drift measurement

$$\text{RateCompFactor} = [T1_{(2)} - T1_{(1)}] / [(T2_{(2)} - \text{Delay}(2)) - (T2_{(1)} - \text{Delay}(1))] \quad (18)$$

where

<i>RateCompFactor</i>	is the rate compensation factor
<i>T1(1)</i>	is the start time of first sync message from Master
<i>T1(2)</i>	is the start time of second sync message from Master
<i>T2(1)</i>	is the arrival time of first sync message on the Bridge
<i>T2(2)</i>	is the arrival time of second sync message on the Bridge
<i>Delay(1)</i>	is the time difference of the first sync message from the start on Master and arrival time on the Bridge
<i>Delay(2)</i>	is the time difference of the second sync message from the start on Master and arrival time on the Bridge

The PTCP transparent clock use the rate compensation factor to correct measured residential times.

$$\text{CompBridgeDelay} = \text{RateCompFactor} * \text{BridgeDelay} \quad (19)$$

where

<i>CompBridgeDelay</i>	is the
<i>RateCompFactor</i>	is the rate compensation factor
<i>BridgeDelay</i>	is the delay of the bridge

6.3.2.3.9.3 Multiple synchronization

PTCP shall be used for clock, time and other kind of synchronization and supports up to 32 synchronization domains. There can be various sources to which all other clocks synchronize. The subdomains are not synchronized at PTCP. Each PTCP slave shall synchronize itself. The delay measurement shall be done for each communication path. The support of clock and time synchronization is are mandatory. Figure 6 shows multiple synchronization.

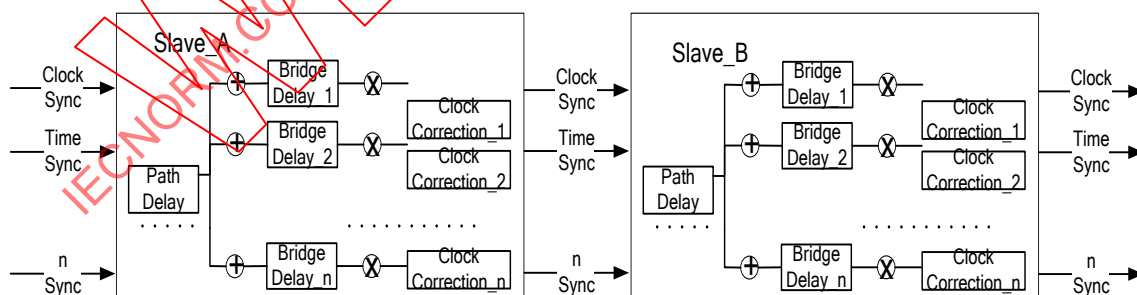


Figure 6 – Multiple synchronization

The Clock Correction consists of frequency offset and drift error of the oscillators. Each Transparent Clock shall correct the residence time of the sync and delay messages with the corresponding value of Clock Correction.

6.3.2.3.9.4 Best master algorithm

The Best Master Algorithm (BMA) specifies an algorithm how a clock determines the preferred clock issuing sync messages. The algorithm runs independently on each clock (master/slave) in a PTCP system. Each clock makes its own decision for the best master.

In computing the best master a clock shall:

- Consider the quality of the most recent sync messages received.
- Include the results of the previous best master computing. However if there are more recent qualified sync messages, the values from that messages shall be considered.

For analyzing the quality of sync message a clock shall consider:

- The master issuing the sync message belongs to the own subdomain.
- Receiving the sync messages periodically.
- The stratum of the master clock.
- The variance which describes the characteristic of the master clock.

If stratum and variance of different masters are equal the BMA use the Source-MAC-Address to decide for the preferred clock (smallest one wins).

6.3.2.4 Media Redundancy support for synchronization with PTCP

To improve the reliability of a peer-to-peer network redundant communication paths are necessary. Redundant communication paths can lead to circulating frames. To avoid circulating frames the redundant communication paths are not used for data messages.

PTCP sync messages use the redundant communication paths to be independent from the reconfiguration of redundant networks and for short communication paths.

The following forwarding algorithm shall be used to avoid circulating sync messages in a network with redundant communication paths. The forwarding algorithm for PTCP takes care that on each communication path only one sync message will be transmitted.

The first received sync message is called primary Sync message. The primary Sync message shall be used for synchronization.

6.3.2.5 Invocation of the PTCP object

For the invocation of the PTCP object the following rules apply:

- PTCP objects shall exist for each Sync ID.

6.3.3 Media redundancy ASE

6.3.3.1 Overview

For media redundancy the IEC 62439-2 applies. The support of the MRP Management Information Base (MIB) is optional.

The Media redundancy ASE specifies a uniform method of achieving media redundancy with deterministic recovery time in case of a single network failure (link failure, switch node failure causing link failures).

NOTE 1 Media redundancy operates transparent to the application. No changes in the application are required.

The Media Redundancy Protocol (MRP) is designed to react deterministic on a single failure of a trunk link or node in the network.

A compliant network shall have a ring topology with multiple nodes.

One of the nodes has the role of a media redundancy manager (MRM). The function of the MRM is to observe and to control the ring topology in order to react on network faults. MRM does this by sending frames on one ring port over the ring and receives them from the ring over its other ring port, and vice-versa in the other direction.

The other nodes in the ring have the role of media redundancy clients (MRC). An MRC reacts on reconfiguration frames from the MRM and can signal link changes on its ring ports.

A compliant node shall have the ability to become MRM, or MRC, or both. Each node requires an integrated switch with at least two ports (ring ports) connected to the ring.

Each node in the ring shall be able to detect the failure or recovery of a trunk link or the failure or recovery of a neighbor node.

The MRP consists of a service and a protocol entity. Both are described by IEC 62439-2.

6.3.3.2 Media redundancy class specification

6.3.3.2.1 General

The Media redundancy ASE defines one object type.

6.3.3.2.2 Template

A Media redundancy object is described by the following template:

ASE:	Media redundancy ASE
CLASS:	Media redundancy
CLASS ID:	not used
PARENT CLASS:	IEEE 802.3 IEEE 802.1D IEEE 802.1 AB
ATTRIBUTES:	
1	(m) Key Attribute: MRP Domain UUID
2	(m) Attribute: MRP Domain Name
3	(m) Attribute: MRP Version
4	(m) Attribute: MRP Role (MANAGER, CLIENT)
5	(m) Attribute: MRP Params
5.1	(c) Constraint: MRP Role = MANAGER
5.1.1	(m) Attribute: Manager Priority
5.1.2	(m) Attribute: Topology Change Interval
5.1.3	(m) Attribute: Topology Change Repeat Count
5.1.4	(m) Attribute: Short Test Interval
5.1.5	(m) Attribute: Default Test Interval
5.1.6	(m) Attribute: Test Monitoring Count
5.2	(c) Constraint: MRP Role = CLIENT
5.2.1	(m) Attribute: Link Down Interval
5.2.2	(m) Attribute: Link Up Interval
5.2.3	(m) Attribute: Link Change Count
6	(m) Attribute: MRP Data Check
6.1	(m) Attribute: MRP Domain UUID
6.2	(m) Attribute: MRP Check
6.2.1	(m) Attribute: Multiple Manager Check
6.2.2	(m) Attribute: Domain UUID Check
7	(m) Attribute: Ring State Data
8	(m) Attribute: Ring Port 1 ID

9	(m) Attribute:	Ring Port 2 ID
10	(o) Attribute:	VLAN ID

SERVICES:

none

6.3.3.2.3 Attributes

MRP Domain UUID

This key attribute defines the redundancy domain representing the ring the Media redundancy object belongs to. It is set to default domain ID or provided as unique ID by engineering.

Attribute type: UUID

MRP Domain Name

This attribute defines the redundancy domain representing the ring the Media redundancy object belongs to. It is set to default domain name or provided as unique ID by engineering.

Attribute type: see Chassis ID

MRP Version

This attribute specifies the version of the Media redundancy object in the redundancy domain referenced by the value of the attribute Domain ID.

Attribute type: Unsigned16

Allowed Values: 1

MRP Role

This attribute specifies the role of the Media redundancy object in the redundancy domain referenced by the value of the attribute Domain ID.

Attribute type: Unsigned16

Allowed Values: MANAGER, CLIENT

MRP Params

This attribute specifies the parameters of the Media redundancy object in the redundancy domain referenced by the value of the attribute Domain ID.

It consists of the following attributes:

MRP Prio

This attribute is present when the value of the MRP role attribute is "MANAGER". This attribute contains the manager's priority. A lower value indicates a higher priority, 0x0000 (highest priority) to 0xF000 (lowest priority) in increments of 0x1000.

Attribute type: Unsigned16

MRP Topology Change Interval

This attribute is present when the value of the MRP role attribute is "MANAGER". This attribute contains the interval for sending topology change frames.

Attribute type: Unsigned16

MRP Topology Change Repeat Count

This attribute is present when the value of the MRP role attribute is "MANAGER". This attribute contains the interval count which controls repeated transmission of topology change frames.

Attribute type: Unsigned16

MRP Short Test Interval

This attribute is present when the value of the MRP role attribute is “MANAGER”. This attribute contains the short interval for sending test frames on ring ports after link changes in the ring.

Attribute type: Unsigned16

MRP Default Test Interval

This attribute is present when the value of the MRP role attribute is “MANAGER”. This attribute contains the default interval for sending test frames on ring ports.

Attribute type: Unsigned16

MRP Test Monitoring Count

This attribute is present when the value of the MRP role attribute is “MANAGER”. This attribute contains the interval count for monitoring the reception of test frames.

Attribute type: Unsigned16

MRP Link Down Interval

This attribute is present when the value of the MRP role attribute is “CLIENT”. This attribute contains the interval for sending link down Link Change frames on ring ports.

Attribute type: Unsigned16

MRP Link Up Interval

This attribute is present when the value of the MRP role attribute is “CLIENT”. This attribute contains the interval for sending link up Link Change frames on ring ports.

Attribute type: Unsigned16

MRP Link Change Count

This attribute is present when the value of the MRP role attribute is “CLIENT”. This attribute contains the Link Change frame count which controls repeated transmission of Link up or link down frames.

Attribute type: Unsigned16

MRP Data Check

This parameter contains the checking rules. It consists of the following attributes:

MRP Domain UUID

This parameter contains the MRP Domain UUID to check against, if the Domain UUID Check attribute is set to TRUE.

Attribute type: UUID

MRP Check

This attribute consists of the following elements:

Multiple Manager Check

This attribute contains the value TRUE to enable the continued evaluation of multiple MRP managers with the reception of MRP-PDUs with the MRP_Type=MRP_Test. Otherwise it shall be set to FALSE.

Attribute type: Boolean

Domain UUID Check

This attribute contains the value TRUE to enable the continued evaluation of MRP Domain UUID with each Remote System Data Change indication as specified within the Neighborhood Check state machine. Otherwise it shall be set to FALSE.

Attribute type: Boolean

Ring State Data

This attribute contains the actual ring state of the Media redundancy object in the redundancy domain.

Attribute type: Unsigned16

Allowed Values:

- OPEN (Ring is open due to link or device failure in ring),
- CLOSED (Ring is closed (normal operation, no error)),
- UNDEFINED (Shall be set if the attribute Real Role State contains the value CLIENT (i.e. Media redundancy object was reconfigured to client role))

Ring Port 1 ID

This attribute specifies one port of a bridge which is assigned as ring port 1 in the redundancy domain referenced by the value of the attribute Domain ID. One port shall only belong to a single redundancy domain.

Attribute type: Unsigned16

Ring Port 2 ID

This attribute specifies another port of a bridge different from Ring Port 1 ID which is assigned as ring port 2 in the redundancy domain referenced by the value of the attribute Domain ID.

Attribute type: Unsigned16

VLAN ID

This optional attribute may be used by the Media redundancy object and specifies its VLAN identifier in the redundancy domain.

Attribute type: Unsigned16

6.3.3.3 Media redundancy service specification

The media redundancy service specification is defined in IEC 62439-2.

6.3.3.4 Media redundancy behavior with diagnosis and alarms

6.3.3.4.1 General

Media redundancy events shall cause diagnosis events and alarm notifications if an application relationship is established and the attribute Check Media Redundancy has the value TRUE.

6.3.3.4.2 Media redundancy diagnosis dependencies

Figure 7 shows the dependencies of media redundancy diagnosis:

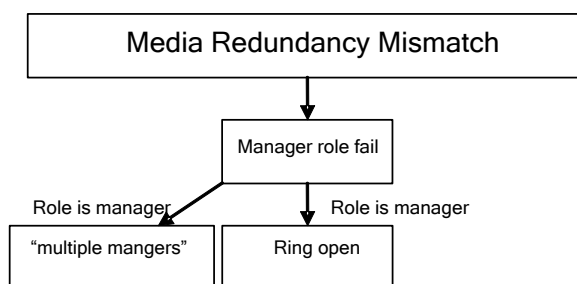


Figure 7 – Media redundancy diagnosis dependencies

If a node in a redundancy domain is not in manager role, a “Manager role fail” diagnosis event shall be signaled. All other media redundancy diagnosis events are locked.

If a node in a redundancy domain is in manager role and another manager is detected, the “multiple managers” event shall be signaled. This event may occur in parallel to ring state event “ring open”.

If a node in a redundancy domain is in manager role and an open ring is detected, the “ring open” event shall be signaled.

These events are indicated by means of the State Change service.

In addition, media redundancy handles the following event for both ring ports of a redundancy domain by means of the Neighborhood Change service:

If the attribute Check Ring Port Neighborhood is TRUE and the attribute Ring Port Domain State is BAD or UNDEFINED, the “Peer Domain ID mismatch” event shall be signaled.

6.3.4 Real time cyclic ASE

6.3.4.1 Overview

The real time protocol cyclic (RTC) is a concept of reading and writing cyclic data by means of buffered services. It uses IEEE 802.3 or User Datagram Protocol (UDP) services.

6.3.4.2 RTC class specification

6.3.4.2.1 General

The RTC ASE defines for each particular CR one RTC object.

6.3.4.2.2 Template

A RTC object is described by the following template:

ASE: RTC ASE

CLASS: RTC

CLASS ID: not used

PARENT CLASS: TOP

ATTRIBUTES:

1 (m) Key Attribute: CREP

2 (m) Attribute: Protocol machine Parameter

2.1 (m) Attribute: DA

2.2	(m) Attribute:	SA
2.3	(m) Attribute:	Frame ID
2.4	(m) Attribute:	Protocol Machine
2.5	(c) Constraint:	Protocol Machine = PPM
2.5.1	(m) Attribute:	VLAN Prio
2.5.2	(m) Attribute:	VLAN ID
2.5.3	(m) Attribute:	Tx Option
2.5.3.1	(m) Attribute:	RT Class
2.5.3.2	(c) Constraint:	RT Class = RT_CLASS_3
2.5.3.2.1	(m) Attribute:	Frame Send Offset
2.5.3.2.2	(m) Attribute:	Tx Port
2.5.3.2.3	(m) Attribute:	Startup Mode
2.5.3.2.4	(m) Attribute:	Media Redundancy
2.5.3.2.5	(c) Constraint:	Media Redundancy = TRUE
2.5.3.2.5.1	(m) Attribute:	Redundant Frame ID
2.5.3.2.5.2	(m) Attribute:	Redundant Frame Send Offset
2.5.3.2.5.3	(m) Attribute:	Redundant Tx Port
2.5.3.2.6	(m) Attribute:	DFF
2.5.3.2.7	(c) Constraint:	DFF = TRUE
2.5.3.2.7.3	(m) Attribute:	Sub Frame Position
2.5.4	(m) Attribute:	Reduction Ratio
2.5.5	(m) Attribute:	Phase
2.5.6	(m) Attribute:	Sequence
2.6	(c) Constraint:	Protocol Machine = CPM
2.6.1	(m) Attribute:	Rx Option
2.6.1.1	(m) Attribute:	RT Class
2.6.1.2	(c) Constraint:	RT Class = RT_CLASS_3
2.6.1.2.1	(m) Attribute:	Startup Mode
2.6.1.2.2	(m) Attribute:	Media Redundancy
2.6.1.2.3	(c) Constraint:	Media Redundancy = TRUE
2.6.1.2.3.1	(m) Attribute:	Redundant Frame ID
2.6.1.2.3.2	(m) Attribute:	Redundancy Watchdog
2.6.1.2.4	(m) Attribute:	DFF
2.6.1.2.5	(c) Constraint:	DFF = TRUE
2.6.1.2.5.1	(m) Attribute:	Sub Frame Position
2.6.2	(m) Attribute:	Data Hold Factor

SERVICES:

1	(m) OpsService:	PPM Set Prov Data
2	(m) OpsService:	PPM Set Prov Status
3	(m) OpsService:	PPM Activate
4	(m) OpsService:	PPM Close
5	(m) OpsService:	PPM Start
6	(m) OpsService:	PPM Error
7	(m) OpsService:	CPM Get Cons Data
8	(m) OpsService:	CPM Get Cons Status
9	(m) OpsService:	CPM Set RedRole
10	(m) OpsService:	CPM Activate
11	(m) OpsService:	CPM NoData
12	(m) OpsService:	CPM Stop

6.3.4.2.3 Attributes

CREP

This key attribute contains the CR's identifier within the CRL class.

Attribute type: Unsigned32

Protocol machine Parameter

This attribute contains the following attributes to describe the behavior of the protocol machines:

DA

This parameter contains the destination address (i.e. the consumer's address) according to IEEE 802.3.

Attribute type: OctetString[6]

SA

This parameter contains the source address (i.e. the provider's address) according to IEEE 802.3.

Attribute type: OctetString[6]

Frame ID

This attribute contains the identifier of the frame.

Attribute type: Unsigned16

Allowed Values: see Table 276

Protocol Machine

This attribute contains the particular protocol machine selected for this CR. The protocol machine is set to PPM (cyclic provider protocol machine), when the device is provider of the particular CR. Otherwise it is set to CPM (cyclic consumer protocol machine), when the device is consumer of the particular CR.

Attribute type: Unsigned8

Allowed Value: PPM, CPM

VLAN Prio

This conditional attribute is present when the value of the protocol machine attribute is "PPM". It contains the priority of the frame according to IEEE 802.1Q.

Attribute type: Unsigned8

VLAN ID

This conditional attribute is present when the value of the protocol machine attribute is "PPM". It contains the VLAN ID according to IEEE 802.1Q.

Attribute type: Unsigned16

Allowed Value: 0

Tx Option

This conditional attribute is present when the value of the protocol machine attribute is "PPM". It specifies the provider's transmit options and consists of the following attributes:

RT Class

This attribute contains the provider's communication class.

Attribute type: Unsigned32

Allowed Values: RT_CLASS_1, RT_CLASS_2, RT_CLASS_3, RT_CLASS_UDP

NOTE 1 RT_CLASS_2 is usable in the context of legacy startup only.

Frame Send Offset

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It contains the relative send offset to the start of the related reduction ratio and phase. The time base is 1 ns. The value shall be less than the duration of one phase.

Attribute type: Unsigned32

Allowed Values: BEST_EFFORT – (mandatory) as soon as possible, 0 – 0x003D08FF for optimized scheduling

Tx Port

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the transmit port, where the frame shall be sent.

Attribute type: Unsigned8

Allowed values: 0x01 to 0xFF

Startup Mode

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the startup mode for RT_CLASS_3.

Allowed values: Legacy, Advanced

Media Redundancy

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of Media Redundancy with Planned Duplication of frames (MRPD).

Allowed values: FALSE, TRUE

Redundant Frame ID

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the identifier of the redundant frame.

Attribute type: Unsigned16

Allowed Values: see Table 276

Redundant Frame Send Offset

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the relative send offset to the start of the related reduction ratio and phase. The time base is 1 ns. The value shall be less than the duration of one phase.

Attribute type: Unsigned32

Allowed Values: BEST_EFFORT – (mandatory) as soon as possible, 0 – 0x003D08FF for optimized scheduling

Redundant Tx Port

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It indicates the port, where the redundant frame shall be sent.

Attribute type: Unsigned8

Allowed values: 0x01 to 0xFF

DFP

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of DFP.

Allowed values: FALSE, TRUE

Sub Frame Position

This conditional attribute is present when the value of the DFP attribute is "TRUE". It defines the position number of the sub frame. The position number shall be strictly increasing within the list of Sub frame Data.

Attribute type: Unsigned8

Allowed values: 0x01 to 0x3F for packetizing and unpacking of sub frames during transport and end to end; 0x40 to 0x7F for packetizing and unpacking of sub frames end to end; 0x00 and 0x80 to 0xFF reserved

Reduction Ratio

This conditional attribute is present when the value of the protocol machine attribute is "PPM". It contains the reduction ratio of the CR. The actual send cycle is determined by Reduction Ratio multiplied with the send clock.

Attribute type: Unsigned16

Allowed Values: see Table 277

Phase

This conditional attribute is present when the value of the protocol machine attribute is "PPM". It indicates in which particular send cycle the data has to be sent. This parameter depends from the parameter Reduction Ratio.

Attribute type: Unsigned16

Allowed values: 1 to selected Reduction Ratio.

Sequence

This conditional attribute is present when the value of the protocol machine attribute is "PPM". It contains the position of the data frame in the sending queue. The value 0 shall be used to indicate an undefined sequence.

Attribute type: Unsigned16

Rx Option

This conditional attribute is present when the value of the protocol machine attribute is "CPM". It specifies the consumer's receive options and consists of the following attributes:

RT Class

This attribute contains the consumer's communication class.

Attribute type: Unsigned32

Allowed Values: RT_CLASS_1, RT_CLASS_2, RT_CLASS_3, RT_CLASS_UDP

NOTE 2 RT_CLASS_2 is usable in the context of legacy startup only.

Startup Mode

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the startup mode of the related AR.

Allowed values: Legacy, Advanced

Media Redundancy

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of MRPD.

Allowed values: FALSE, TRUE

Redundant Frame ID

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the identifier of the redundant frame.

Attribute type: Unsigned16

Allowed Values: see Table 276

Redundancy Watchdog Timer

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the value for the redundancy watchdog timer.

Attribute type: Unsigned16

Time Base: 1 s

Allowed values: 0x0001 to 0x0100

Default value: 0x0001

NOTE 3 The redundancy watchdog is meant to detect "static" ring problems (e.g. a cable is pulled). Thus the resolution of the redundancy watchdog timer is relaxed.

DFP

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of DFP.

Allowed values: FALSE, TRUE

Sub Frame Position

This conditional attribute is present when the value of the DFP attribute is "TRUE". It defines the position number of the sub frame. The position number shall be strictly increasing within the list of Sub frame Data.

Attribute type: Unsigned8

Allowed values: 0x01 to 0x3F for packetizing and unpacking of sub frames during transport and end to end; 0x40 to 0x7F for packetizing and unpacking of sub frames end to end; 0x00 and 0x80 to 0xFF reserved

Data Hold Factor

This conditional attribute is present when the value of the protocol machine attribute is "CPM". It contains a factor, which is used to calculate the Data Hold Time. This time is the time interval for holding last output data if no IO data have been received.

The interval for the Data Hold Time is the product of the attribute values of Data Hold Factor, Send Clock Factor, Reduction Ratio and the time base of 31,25 µs related to the monitored consumer. The Data Hold Time shall equal or less 1,92 s if the value of the AR attribute RT Class is 0x01, 0x02 or 0x03 and equal or less 49,152 s if the value of the AR attribute RT Class is 0x04.

Attribute type: Unsigned16

Allowed values: 0x0001 to 0x1E00 for RT Class 0x01, 0x02 or 0x03; 0x0001 to 0xF000 for RT Class 0x04

6.3.4.3 RTC service specification

6.3.4.3.1 PPM Set Prov Data

This local service is used to update the provider's transmit buffer with new data. Table 50 shows the parameters of the service.

Table 50 – PPM Set Prov Data

Parameter name	req	cnf
Argument	M	
CREP	M	
Data	M	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Data

This parameter contains the data to update the transmit buffer.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTEXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.2 PPM Set Prov Status

This local service is used to update the APDU status of the provider's transmit buffer. Table 51 shows the parameters of the service.

Table 51 – PPM Set Prov Status

Parameter name	req	cnf
Argument	M	
CREP	M	
Status	M	

Parameter name	req	cnf
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Status

This parameter contains the APDU status to update for the transmit buffer.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.3 PPM Activate

This local service is used to activate the provider protocol machine (PPM). Table 52 shows the parameters of the service.

Table 52 – PPM Activate

Parameter name	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
VLAN Prio	M	
VLAN ID	M	
Tx Option	M	
RT Class	M	

Parameter name	req	cnf
Frame Send Offset	C	
Tx Port	C	
Startup Mode	C	
Media Redundancy	C	
Redundant Frame ID	C	
Redundant Frame Offset	C	
Redundant Frame Tx Port	C	
DFP	C	
Sub Frame Position	C	
Reduction Ratio	M	
Phase	M	
Sequence	M	
Default Values	M	
Default Status	M	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

DA

This parameter contains the destination MAC address to be used for the RTC frame.

SA

This parameter contains the source MAC address to be used for the RTC frame.

Frame ID

This parameter contains the Frame ID to be used for the RTC frame.

VLAN Prio

This parameter contains the VLAN Prio to be used for the RTC frame.

VLAN ID

This parameter contains the VLAN ID to be used for the RTC frame.

Tx Option

This attribute defines the transmit option. It is composed of the following elements:

RT Class

This attribute contains the provider's communication class.

Frame Send Offset

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It contains the relative send offset to the start of the related reduction ratio and phase.

Tx Port

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the transmit port, where the frame shall be sent.

Startup Mode

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the startup mode of the related AR.

Media Redundancy

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of MRPD.

Redundant Frame ID

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the identifier of the redundant frame.

Redundant Frame Send Offset

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the relative send offset to the start of the related reduction ratio and phase.

Redundant Tx Port

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It indicates the port, where the redundant frame shall be sent.

DFP

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of DFP.

Sub Frame Position

This conditional attribute is present when the value of the DFP attribute is "TRUE". It defines the position number of the sub frame.

Reduction Ratio

This parameter contains the Reduction Ratio to be used for the RTC frame.

Phase

This parameter contains the Phase to be used for the RTC frame.

Sequence

This parameter contains the Sequence to be used for the RTC frame.

Default Value

This parameter contains the initial data for the transmit buffer.

Default Status

This parameter contains the initial APDU status for the transmit buffer.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.4 PPM Close

This local service is used to close the provider protocol machine (PPM) and thus to stop the transmission. Table 53 shows the parameters of the service.

Table 53 – PPM Close

Parameter name	req	cnf
Argument	M	
CREP	M	
Result(+)		M
CREP		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Result(+)

This parameter indicates that the service request succeeded.

6.3.4.3.5 PPM Start

This local service is used to start the transmission of the provider protocol machine (PPM). Table 54 shows the parameters of the service.

Table 54 – PPM Start

Parameter name	ind
Argument	M
CREP	M

Argument

The argument shall convey the service specific parameters of the service indication.

CREP

This parameter contains the CR's local identifier.

6.3.4.3.6 PPM Error

This local service is used to indicate an error during transmission of the provider protocol machine (PPM). Table 55 shows the parameters of the service.

Table 55 – PPM Error

Parameter name	ind
Argument	M
CREP	M
No Data Send	M

Argument

The argument shall convey the service specific parameters of the service indication.

CREP

This parameter contains the CR's local identifier.

No Data Send

This parameter indicates the failure of data transmission.

6.3.4.3.7 CPM Get Cons Data

This local service is used to read data from a consumer's receive buffer. Table 56 shows the parameters of the service.

Table 56 – Get Cons Data

Parameter name	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M
Data		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Result(+)

This parameter indicates that the service request succeeded.

Data

This parameter contains the data from the receive buffer.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.8 CPM Get cons status

This local service is used to read the APDU status from the consumer's receive buffer. Table 57 shows the parameters of the service.

Table 57 – CPM Get cons status

Parameter name	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M
Status		M
Recv Counter		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Result(+)

This parameter indicates that the service request succeeded.

Status

This parameter contains the APDU status.

Recv Counter

This parameter contains the receive counter.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.9 CPM Set RedRole

This local service is used to set the redundancy status of the CR. Table 58 shows the parameters of the service.

Table 58 – CPM Set RedRole

Parameter name	req	cnf
Argument	M	
CREP	M	
Red Role	M	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Red Role

This parameter determines the redundancy role (primary, backup) of the CR.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.10 CPM Activate

This local service is used to activate the consumer protocol machine (CPM). Table 59 shows the parameters of the service.

Table 59 – CPM Activate

Parameter name	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	

Parameter name	req	cnf
Frame ID	M	
Rx Option	M	
RT Class	M	
Startup Mode	C	
Media Redundancy	C	
Redundant Frame ID	C	
Redundant Watchdog Timer	C	
DFP	C	
Sub Frame Position	C	
Expected Length	M	
Start Ton	M	
Datahold Factor	M	
Default Value	M	
Default Status	M	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

DA

This parameter shall be used for the destination MAC address.

SA

This parameter shall be used for the source MAC address.

Frame ID

This parameter contains the value of the corresponding attribute of the CR ASE object.

Rx Option

This conditional attribute specifies the consumer's receive options and consists of the following attributes:

RT Class

This attribute contains the consumer's communication class.

Startup Mode

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the startup mode of the related AR.

Media Redundancy

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of MRPD.

Redundant Frame ID

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the identifier of the redundant frame.

Redundancy Watchdog Timer

This conditional attribute is present when the value of the Media Redundancy attribute is set to "TRUE". It contains the value for the redundancy watchdog timer.

DFP

This conditional attribute is present when the value of the RT Class attribute is "RT_CLASS_3". It indicates the usage of DFP.

Sub Frame Position

This conditional attribute is present when the value of the DFP attribute is "TRUE". It defines the position number of the sub frame.

Expected Length

This parameter contains the expected length of the CR.

Start Ton

This parameter shall be set to true if the timeout shall be monitored immediately. Otherwise it starts with the reception of the first frame.

Datahold Factor

This parameter contains the datahold factor.

Default Value

This parameter contains the initial value of the receive buffer's data.

Default Status

This parameter contains the initial value of the receive buffer's APDU status.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned8

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.4.3.11 CPM NoData

This local service is used to indicate that the consumer protocol machine (CPM) did not receive data within the data hold timer. Table 60 shows the parameters of the service.

Table 60 – CPM NoData

Parameter name	req	cnf
Argument	M	
CREP	M	

Parameter name	req	cnf
Result(+)		M
CREP		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Result(+)

This parameter indicates that the service request succeeded.

6.3.4.3.12 CPM Stop

This local service is used to stop the consumer protocol machine (CPM) and thus the reception. Table 61 shows the parameters of the service.

Table 61 – CPM Stop

Parameter name	req	cnf
Argument	M	
CREP	M	
Result(+)		M
CREP		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter contains the CR's local identifier.

Result(+)

This parameter indicates that the service request succeeded.

6.3.5 Real time acyclic ASE**6.3.5.1 Overview**

The real time protocol acyclic (RTA) is a concept of reading and writing acyclic data by means of acknowledged services. It uses IEEE 802.3 or UDP services.

6.3.5.2 RTA class specification**6.3.5.2.1 General**

The RTA ASE defines for each CR one RTA object type:

6.3.5.2.2 Template

A RTA object is described by the following template:

ASE: RTA ASE
CLASS: RTA
CLASS ID: not used
PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Key Attribute: CREP
2	(m) Attribute: Protocol machine Parameter
2.1	(m) Attribute: DA
2.2	(m) Attribute: SA
2.3	(m) Attribute: Frame ID
2.4	(m) Attribute: VLAN Prio
2.5	(m) Attribute: VLAN ID
2.6	(m) Attribute: RTA Timeout Factor
2.7	(m) Attribute: M Retry
2.8	(m) Attribute: Send Seq Num
2.9	(m) Attribute: Ack Seq Num
2.10	(m) Attribute: Dst IP
2.11	(m) Attribute: Src IP
2.12	(m) Attribute: Transport

SERVICES:

1	(m) OpsService: APMS Activate
2	(m) OpsService: APMR Activate
3	(m) OpsService: APMS A Data
4	(m) OpsService: APMR A Data
5	(m) OpsService: APMR Ack
6	(m) OpsService: APMS Error
7	(m) OpsService: APMR Error
8	(m) OpsService: APMS Close
9	(m) OpsService: APMR Close

6.3.5.2.3 Attributes

CREP

This key attribute identifies the instance of the protocol machine to which the data belong.

Attribute type: Unsigned32

Protocol machine Parameter

This attribute contains the following attributes to describe the behavior of the protocol machines:

DA

This attribute contains the physical address of the destination device according to IEEE 802

Attribute type: OctetString[6]

SA

This attribute contains the physical address of the device according to IEEE 802.3

Attribute type: OctetString[6]

Frame ID

This attribute contains the identifier of the data within the frame.

Attribute type: Unsigned16

Allowed Values: ALARM_HIGH, ALARM_LOW

VLAN Prio

This attribute contains the priority of the frame according to IEEE 802.1Q.

Attribute type: Unsigned8

Allowed Values: LOW, HIGH

VLAN ID

This attribute contains the VLAN ID according to IEEE 802.1Q.

Attribute type: Unsigned16

Allowed Value: 0

RTA Timeout Factor

This attribute contains the timeout factor within a sent RTA Data frame has to be Acknowledged (ACK).

Attribute type: Unsigned16

Time Base: 100 ms

Allowed Values: 1 up to 100 mandatory, 101 to $2^{16}-1$ optional

M Retry

This attribute contains the number of retries if no ACK received within RTA Timeout Factor. The value range is from 3 to 15 retries.

Attribute type: Unsigned16

Send Seq Num

This attribute contains the Sender Sequence Number.

Attribute type: Unsigned16

Allowed Values: 0 to 0x7FFF, 0xFFFF, 0xFFFE

Ack Seq Num

This attribute contains the Acknowledge Sequence Number.

Attribute type: Unsigned16

Allowed Values: 0 to 0x7FFF, 0xFFFF, 0xFFFE

Dst IP

This attribute contains the destination IP address if UDP Services shall be used.

Attribute type: Unsigned32

Src IP

This attribute contains the local IP address if UDP Services shall be used.

Attribute type: Unsigned32

Transport

This attribute contains the requested conveyance.

Attribute type: Unsigned8

Allowed Values: RTC, UDP

6.3.5.3 RTA service specification

6.3.5.3.1 APMS Activate

This local service shall be used to activate the Acyclic Protocol Machine Sender (APMS). Table 62 shows the parameters of the service.

Table 62 – APMS Activate

Parameter name	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
VLAN Prio	M	
VLAN ID	M	
RTA Timeout Factor	M	
M Retry	M	
Transport	M	
Dst IP	O	
Src IP	O	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request. These parameters shall correspond with the ASE attributes.

CREP

This parameter identifies the instance of the protocol machine.

DA

This parameter shall be used for the physical address of the destination device according to IEEE 802.3.

SA

This parameter shall be used for the physical address of the device according to IEEE 802.3.

Frame ID

This parameter shall be used for the Frame ID.

VLAN Prio

This parameter shall contain the priority of the frame according to IEEE 802.1Q

VLAN ID

This parameter shall contain the VLAN ID according to IEEE 802.1Q.

RTA Timeout Factor

This parameter shall contain the timeout factor.

M Retry

This parameter shall contain the number of retries.

Dst IP

This parameter shall contain the destination IP address if UDP services shall be used. Not used if IEEE 802.3 services shall be used.

Src IP

This parameter shall contain the local IP address if UDP services shall be used. Not used if IEEE 802.3 services shall be used.

Transport

This parameter shall contain the requested conveyance.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned16

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.5.3.2 APMR Activate

This local service shall be used to activate the Acyclic Protocol Machine Receiver (APMR).

Table 63 shows the parameters of the service.

Table 63 – APMR Activate

Parameter name	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
VLAN Prio	M	
VLAN ID	M	
Transport	M	
Dst IP	O	
Src IP	O	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
Error Class		M
Error code		M

Argument

The argument shall convey the service specific parameters of the service request. These parameters shall correspond with the ASE attributes.

CREP

This parameter identifies the instance of the protocol machine.

DA

This parameter shall be used for the physical address of the destination device according to IEEE 802.3.

SA

This parameter shall be used for the physical address of the device according to IEEE 802.3.

Frame ID

This parameter shall be used for the Frame ID.

VLAN Prio

This parameter shall contain the priority of the frame according to IEEE 802.1Q

VLAN ID

This parameter shall contain the VLAN ID according to IEEE 802.1Q.

Transport

This parameter shall contain the requested conveyance.

Dst IP

This parameter shall contain the destination IP address if UDP services shall be used. Not used if IEEE 802.3 services shall be used.

Src IP

This parameter shall contain the local IP address if UDP services shall be used. Not used if IEEE 802.3 services shall be used.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Class

This parameter selects the error class.

Type: Unsigned16

Allowed Value: CTXT

Error code

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.5.3.3 APMS A Data

This service shall be used to send RTA Data. After a RTA Data frame was sent, it has to be acknowledged within a specified time (based on RTA Timeout Factor). If no acknowledge (ACK) arrives the data frame will be resend a specified number of times (M Retry). If no ACK arrived at all a APMS Error will be indicated (TIMEOUT). The communication uses sequence numbers to ensure proper sequence of data. Both communication endpoints have to be setup for communication to ensure proper sequence number setup. Only one service can be handled at a time. So the service has to be finished before a new one can be issued (INVALID_STATE result shall be returned if not finished). Table 64 shows the parameters of the service.

Table 64 – APMS A Data

Parameter name	req	cnf
Argument	M	
CREP	M	
Data	M	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
ERRCLS		M
ERRCODE		M

Argument

The argument shall convey the service specific parameters of the service request. These parameters shall correspond with the ASE attributes.

CREP

This parameter identifies the instance of the protocol machine.

Data

This parameter shall be used for RTA-SDU data to be send. The length is limited by ADPU size.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

ERRCLS

This parameter selects the error class.

Type: Unsigned16

Allowed Value: CTXT

ERRCODE

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.5.3.4 APMR A Data

This local service may be used to receive RTA data. Table 65 shows the parameters of the service.

Table 65 – APMR A Data

Parameter name	ind
Argument	M
CREP	M
Data	M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter identifies the instance of the protocol machine.

Data

This parameter contains the received RTA-SDU data

Result(+)

This parameter indicates that the service request succeeded.

6.3.5.3.5 APMR Ack

This service shall be used to send a RTA ACK after reception of a RTA Data. After RTA Data was received it has to be acknowledged by sending an ACK. Sending the ACK signals the sender that the Data was successfully received and further data may be send. If the ACK will not be sent, the sender retries and will timeout if no ACK arrives at all. Only one service call can be handled at a time. So the service has to be finished before a new one can be issued (INVALID_STATE result is set in not finished). Table 66 shows the parameters of the service.

Table 66 – APMR Ack

Parameter name	req	cnf

Parameter name	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M
Result(-)		S
CREP		M
ERRCLS		M
ERRCODE		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter identifies the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

ERRCLS

This parameter selects the error class.

Type: Unsigned16

Allowed Value: CTXT

ERRCODE

This parameter selects the error code.

Type: Unsigned16

Allowed Value: INVALID_STATE

6.3.5.3.6 APMS error

This local service may be used to indicate an APMS error. Table 67 shows the parameters of the service.

Table 67 – APMS Error

Parameter name	ind
Argument	M
CREP	M
ERRCLS	M
ERRCODE	M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter identifies the instance of the protocol machine.

ERRCLS

This parameter selects the error class.

Type: Unsigned16

Allowed Value: see Table 68

ERRCODE

This parameter selects the error code.

Type: Unsigned16

Allowed Value: see Table 68

Table 68 – APMS Error ERRCLS/ERRCODE

ERRCLS	ERRCODE	Meaning
PROTOCOL	TIMEOUT	Transmission timeout occurred
PROTOCOL	SEQNUM	Sequence number mismatch
PROTOCOL	LMPM	LMPM error

6.3.5.3.7 APMR Error

This local service may be used to indicate an APMR error. Table 69 shows the parameters of the service.

Table 69 – APMR Error

Parameter name	ind
Argument	M
CREP	M
ERRCLS	M
ERRCODE	M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter identifies the instance of the protocol machine.

ERRCLS

This parameter selects the error class.

Type: Unsigned16

Allowed Value: see Table 70

ERRCODE

This parameter selects the error code.

Type: Unsigned16

Allowed Value: see Table 70

Table 70 – APMR Error ERRCLS/ERRCODE

ERRCLS	ERRCODE	Meaning

ERRCLS	ERRCODE	Meaning
PROTOCOL	SEQNUM	Sequence number of received data mismatch
PROTOCOL	LMPM	LMPM error

6.3.5.3.8 APMS_Close

This local service shall be used to close the Sender protocol machine (APMS). Table 71 shows the parameters of the service.

Table 71 – APMS_Close

Parameter name	req	cnf
Argument	M	
CREP	M	
ErrCode	U	
Result(+)		S
CREP		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter identifies the instance of the protocol machine.

ErrCode

This optional parameter selects the error code.

Type: Unsigned16

Allowed Value: See corresponding ASE attribute

Result(+)

This parameter indicates that the service request succeeded.

6.3.5.3.9 APMR_Close

This local service shall be used to close the Receiver protocol machine (APMR). Table 72 shows the parameters of the service.

Table 72 – APMR_Close

Parameter name	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter identifies the instance of the protocol machine.

Result(+)

This parameter indicates that the service request succeeded.

6.3.6 Remote procedure call ASE

6.3.6.1 Overview

The remote procedure call (RPC) concepts according to Publication C706 of the Open Group shall be applied. This ASE defines the special usage of the Publication C706 of the Open Group as a transport protocol.

6.3.6.2 RPC class specification

6.3.6.2.1 Template

The RPC object is described by the following template:

ASE:	RPC ASE
CLASS:	RPC
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES and SERVICES	
1	(m) Key Attribute: Implicit
2	(m) Attribute: Interface EPMap
2.1	(o) OpsService: Insert
2.2	(o) OpsService: Delete
2.3	(m) OpsService: Lookup
2.4	(o) OpsService: Map
2.5	(m) OpsService: LookupHandleFree
2.6	(o) OpsService: InqObject
2.7	(o) OpsService: MgmtDelete
3	(m) Attribute: Interface IOD, IOC, IOS, IOPParameterServer
3.1	(m) OpsService: Connect
3.2	(m) OpsService: Release
3.3	(m) OpsService: Read
3.4	(m) OpsService: Write
3.5	(m) OpsService: Control

6.3.6.2.2 Attributes

Implicit

The key attribute Implicit indicates that the object is implicitly addressed by the service.

Interface EPMap

This attribute contains the end point mapper interface of the device. There is exactly one instance of this interface per device.

Interface IOD, IOC, IO system, IOPParameterServer

This attribute contains the IO interface of the device. The type of the interface determines the addressed functionality (i.e. IO device, IO controller, IO supervisor or IO parameter server). From each type there may be several instances of each interface; for details see Annex A.

6.3.6.2.3 General service behavior

At any point of time at most one outstanding service call is supported between an RPC client and the RPC server; thus succeeding service calls have to be sequentialized. It is allowed to have calls to different services in parallel (e.g. one outstanding write and read in parallel).

The explicit or implicit abort of a remote procedure call according to Publication C706 of the Open Group is implemented by local means.

The end point mapper is mapped to port 0x8892 in opposite to the end point mapper as specified by Publication C706 which is mapped to port 0x0087 (135). Thus the end point mapper library functions (bound on port 135) of a standard RPC client are not usable; the client shall call the end point mapper interface (e.g. Lookup) at the port 0x8892 and evaluate its response.

The lookup response contains for a particular interface instance in its Protocol Tower information the destination port to use for the communication to this instance.

If a client makes a direct call to an interface instance, it has to respect the port forwarding of RPC end point mapper. The initial client request to the interface is directed to port 0x8892; the response of the server is issued from its port. This source port of the response shall thus be used as destination port in further client calls to the interface instance.

6.3.6.3 RPC service specification

6.3.6.3.1 Interface EMap

6.3.6.3.1.1 Insert

For the definition of this optional service see Publication C706 of the Open Group.

6.3.6.3.1.2 Delete

For the definition of this optional service see Publication C706 of the Open Group.

6.3.6.3.1.3 Lookup

For the definition of this service see Publication C706 of the Open Group.

6.3.6.3.1.4 Map

For the definition of this optional service see Publication C706 of the Open Group.

6.3.6.3.1.5 LookupHandleFree

For the definition of this service see Publication C706 of the Open Group.

6.3.6.3.1.6 InqObject

For the definition of this optional service see Publication C706 of the Open Group.

6.3.6.3.1.7 MgmtDelete

For the definition of this optional service see Publication C706 of the Open Group.

6.3.6.3.2 Interface IOD, IOC, IOS, IOPParameterServer

6.3.6.3.2.1 Connect

This confirmed service shall be used to establish a connection by means of Connectionless Remote Procedure Call (CL-RPC) Table 73 shows the parameters of the service.

Table 73 – Connect

Parameter name	req	ind	rsp	cnf

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result(-)			S	S(=)
PNIO Status			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

Args Max

This parameter shall contain the maximum buffer size available for the response.

Type: Unsigned32

Args Len

This parameter shall contain the actual number of octets for the parameter Args.

Type: Unsigned32

Args

This parameter shall contain the RPC user data.

Type: OctetString

Result(+)

This parameter indicates that the service request succeeded.

PNIO Status

This parameter shall be set to zero.

Type: Unsigned32

Result(-)

This parameter indicates that the service request failed.

PNIO Status

This parameter shall be set according to the current error and shall consist of the elements Error code = "IODConnectRes", Error Decode = "PNIO", Error code 1, and Error code 2. Error code 1 and Error code 2 shall contain the actual error code according to the protocol definition.

Type: Unsigned32

6.3.6.3.2.2 Release

This confirmed service shall be used to release a connection by means of CL-RPC. Table 74 shows the parameters of the service.

Table 74 – Release

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result(-)			S	S(=)
PNIO Status			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

Args Max

This parameter shall contain the maximum buffer size available for the response.

Type: Unsigned32

Args Len

This parameter shall contain the actual number of octets for the parameter Args.

Type: Unsigned32

Args

This parameter shall contain the RPC user data.

Type: OctetString

Result(+)

This parameter indicates that the service request succeeded.

PNIO Status

This parameter shall be set to zero.

Type: Unsigned32

Result(-)

This parameter indicates that the service request failed.

PNIO Status

This parameter shall be set according to the current error and shall consist of the elements Error code = "IODReleaseRes", Error Decode = "PNIO", Error code 1, and Error code 2. Error code 1 and Error code 2 shall contain the actual error code according to the protocol definition.

Type: Unsigned32

6.3.6.3.2.3 Read

This confirmed service shall be used to convey a higher layer read service by means of CL-RPC. Table 75 shows the parameters of the service.

Table 75 – Read

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result(-)			S	S(=)
PNIO Status			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

Args Max

This parameter shall contain the maximum buffer size available for the response.

Type: Unsigned32

Args Len

This parameter shall contain the actual number of octets for the parameter Args.

Type: Unsigned32

Args

This parameter shall contain the RPC user data.

Type: OctetString

Result(+)

This parameter indicates that the service request succeeded.

PNIO Status

This parameter shall be set to zero.

Type: Unsigned32

Result(-)

This parameter indicates that the service request failed.

PNIO Status

This parameter shall be set according to the current error and shall consist of the elements Error code = "IODReadRes", Error Decode = "PNIORW", Error code 1, and Error code 2. Error code 1 and Error code 2 shall contain the actual error code according to the protocol definition.

Type: Unsigned32

6.3.6.3.2.4 Write

This confirmed service shall be used to convey a higher layer write service by means of CL–RPC. Table 76 shows the parameters of the service.

Table 76 – Write

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result(-)			S	S(=)
PNIO Status			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

Args Max

This parameter shall contain the maximum buffer size available for the response.

Type: Unsigned32

Args Len

This parameter shall contain the actual number of octets for the parameter Args.

Type: Unsigned32

Args

This parameter shall contain the RPC user data.

Type: OctetString

Result(+)

This parameter indicates that the service request succeeded.

PNIO Status

This parameter shall be set to zero.

Type: Unsigned32

Result(-)

This parameter indicates that the service request failed.

PNIO Status

This parameter shall be set according to the current error and shall consist of the elements Error code = "IODWriteRes", Error Decode = "PNIORW", Error code 1, and Error code 2. Error code 1 and Error code 2 shall contain the actual error code according to the protocol definition.

Type: Unsigned32

6.3.6.3.2.5 Control

This confirmed service shall be used to convey a higher layer control service by means of CL-RPC. Table 77 shows the parameters of the service.

Table 77 – Control

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result(-)			S	S(=)
PNIO Status			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

Args Max

This parameter shall contain the maximum buffer size available for the response.

Type: Unsigned32

Args Len

This parameter shall contain the actual number of octets for the parameter Args.

Type: Unsigned32

Args

This parameter shall contain the RPC user data.

Type: OctetString

Result(+)

This parameter indicates that the service request succeeded.

PNIO Status

This parameter shall be set to zero.

Type: Unsigned32

Result(-)

This parameter indicates that the service request failed.

PNIO Status

This parameter shall be set according to the current error and shall consist of the elements Error code = "IOXControlRes", Error Decode = "PNIO", Error code 1, and Error code 2. Error code 1 and Error code 2 shall contain the actual error code according to the protocol definition.

Type: Unsigned32

6.3.7 IEEE 802.1AB ASE

6.3.7.1 Overview

The concepts according to IEEE 802.1AB standard shall be applied. This part of the specification defines the special usage of Chassis ID, Port ID, Chassis ID Subtype, Port ID Subtype, and Time to live. Furthermore, extensions for the OUI regarding frames and MIB are defined.

Furthermore, the IETF RFC 3490 (IDNA) shall be applied for the content syntax of Chassis ID and Port ID.

If Simple Network Management Protocol (SNMP) is supported, the system shall conform to the LLDP management specifications in and shall implement the sections of the basic LLDP MIB for the operating mode being implemented. If SNMP is not supported, the system shall provide storage and retrieval capability equivalent to the functionality for the operating mode being implemented.

6.3.7.2 IEEE 802.1AB class specification

6.3.7.2.1 Template

The IEEE 802.1AB object is described by the following template:

ASE:	Link Layer Discovery ASE
CLASS:	IEEE 802.1AB
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: Chassis ID
3	(m) Attribute: Port ID
4	(m) Attribute: Multiple Interface Mode
5	(m) Attribute: LLDP Time To Live
6	(o) Attribute: Port Description
7	(o) Attribute: System Name
8	(o) Attribute: System Description
9	(o) Attribute: System Capabilities
10	(m) Attribute: Management Address
11	(o) Attribute: Object Identifier
12	(o) Attribute: List of Organizationally-specific extensions
12.1	(m) Attribute: Organizationally Unique Identifier
12.2	(c) Constraint: Organizationally Unique Identifier = 00-80-C2
12.2.1	(m) Attribute: List of IEEE 802.1 Subtypes
12.2.1.1	(m) Attribute: Subtype
12.2.1.2	(c) Constraint: Subtype = PORT_VLAN_ID
12.2.1.2.1	(m) Attribute: Port VLAN Identifier
12.2.1.3	(c) Constraint: Subtype = PORT_AND_PROTOCOL_VLAN_ID
12.2.1.3.1	(m) Attribute: Flags
12.2.1.3.2	(m) Attribute: PPVID Reference Number
12.2.1.4	(c) Constraint: Subtype = VLAN_NAME
12.2.1.4.1	(m) Attribute: VLAN ID
12.2.1.4.2	(m) Attribute: VLAN Name
12.2.1.5	(c) Constraint: Subtype = PROTOCOL_IDENTITY
12.2.1.5.1	(m) Attribute: List of Protocol Identities
12.2.1.5.1.1	(m) Attribute: Protocol Identity

12.3	(c) Constraint:	Organizationally Unique Identifier = 00-12-0F
12.3.1	(m) Attribute:	List of IEEE 802.3 Subtypes
12.3.1.1	(m) Attribute:	Subtype
12.3.1.2	(c) Constraint:	Subtype = MAC_PHY_CONFIGURATION_STATUS
12.3.1.2.1	(m) Attribute:	Auto Negotiation Support And Status
12.3.1.2.2	(m) Attribute:	Advertised Capability
12.3.1.2.3	(m) Attribute:	Operational MAU type
12.3.1.3	(c) Constraint:	Subtype = POWER_VIA_MDI
12.3.1.3.1	(m) Attribute:	MDI Power Support
12.3.1.3.2	(m) Attribute:	Power Pair
12.3.1.3.3	(m) Attribute:	Power Class
12.3.1.4	(c) Constraint:	Subtype = LINK_AGGREGATION
12.3.1.4.1	(m) Attribute:	Link Aggregation Status
12.3.1.4.2	(m) Attribute:	Aggregated Port ID
12.3.1.5	(c) Constraint:	Subtype = MAXIMUM_FRAME_SIZE
12.3.1.5.1	(m) Attribute:	Maximum Frame Size
12.4	(c) Constraint:	Organizationally Unique Identifier = 00-0E-CF
12.4.1	(m) Attribute:	List of PNIO Subtypes
12.4.1.1	(m) Attribute:	Subtype
12.4.1.2	(c) Constraint:	Subtype = LLDP_PNIO_DELAY
12.4.1.2.1	(m) Attribute:	PnioDelay
12.4.1.2.1.1	(m) Attribute:	PortRxDelayLocal
12.4.1.2.1.2	(m) Attribute:	PortRxDelayRemote
12.4.1.2.1.3	(m) Attribute:	PortTxDelayLocal
12.4.1.2.1.4	(m) Attribute:	PortTxDelayRemote
12.4.1.2.1.5	(m) Attribute:	CableDelayLocal
12.4.1.3	(c) Constraint:	Subtype = LLDP_PNIO_PORTSTATUS
12.4.1.3.1	(m) Attribute:	PnioPortStatus
12.4.1.3.1.1	(m) Attribute:	RTClass2PortStatus
12.4.1.3.1.2	(m) Attribute:	RTClass3PortStatus
12.4.1.3.1.2.1	(m) Attribute:	Status
12.4.1.3.1.2.2	(m) Attribute:	Mode
12.4.1.4	(c) Constraint:	Subtype = LLDP_PNIO_ALIAS
12.4.1.4.1	(m) Attribute:	PnioAlias
12.4.1.5	(c) Constraint:	Subtype = LLDP_PNIO_MRP_PORT_STATUS
12.4.1.5.1	(m) Attribute:	PnioMRPPortStatus
12.4.1.5.1.1	(m) Attribute:	MRPDomainUUID
12.4.1.6	(c) Constraint:	Subtype = LLDP_PNIO_INTERFACE_MAC_ADDRESS
12.4.1.6.1	(m) Attribute:	PnioInterfaceMacAddress
12.4.1.7	(c) Constraint:	Subtype = LLDP_PNIO_PTCP_STATUS
12.4.1.7.1	(m) Attribute:	PnioPTCPStatus
12.4.1.7.1.1	(m) Attribute:	PnioPTCPMasterSourceAddress
12.4.1.7.1.2	(m) Attribute:	PnioPTCPSubdomainUUID
12.4.1.7.1.3	(m) Attribute:	PnioIRDataUUID
12.4.1.7.1.4	(m) Attribute:	PnioLengthOfPeriod
12.4.1.7.1.4.1	(m) Attribute:	Length
12.4.1.7.1.4.2	(m) Attribute:	Validity
12.4.1.7.1.5	(m) Attribute:	PnioRedPeriodBegin
12.4.1.7.1.6	(m) Attribute:	PnioOrangePeriodBegin
12.4.1.7.1.7	(m) Attribute:	PnioGreenPeriodBegin
12	(m) Attribute:	List of Remote Systems Data
13.1	(m) Attribute:	Port ID
13.2	(m) Attribute:	Chassis ID

13.3	(m) Attribute:	MAC Address
13.4	(m) Attribute:	LLDP Time To Live
13.5	(o) Attribute:	Port Description
13.6	(o) Attribute:	System Name
13.7	(o) Attribute:	System Description
13.8	(o) Attribute:	System Capabilities
13.9	(o) Attribute:	Management Address
13.10	(o) Attribute:	Object Identifier
13.11	(m) Attribute:	Line Delay
13.12	(m) Attribute:	List of Organizationally-specific extensions
13.12.1	(m) Attribute:	Organizationally Unique Identifier
13.12.2	(c) Constraint:	Organizationally Unique Identifier = 00-80-C2
13.12.2.1	(m) Attribute:	List of IEEE 802.1 Subtypes
13.12.2.1.1	(m) Attribute:	Subtype
13.12.2.1.2	(c) Constraint:	Subtype = PORT_VLAN_ID
13.12.2.1.2.1	(m) Attribute:	Port VLAN Identifier
13.12.2.1.3	(c) Constraint:	Subtype = PORT_AND_PROTOCOL_VLAN_ID
13.12.2.1.3.1	(m) Attribute:	Flags
13.12.2.1.3.2	(m) Attribute:	PPVID Reference Number
13.12.2.1.4	(c) Constraint:	Subtype = VLAN_NAME
13.12.2.1.4.1	(m) Attribute:	VLAN ID
13.12.2.1.4.2	(m) Attribute:	VLAN Name
13.12.2.1.5	(c) Constraint:	Subtype = PROTOCOL_IDENTITY
13.12.2.1.5.1	(m) Attribute:	List of Protocol Identities
13.12.2.1.5.1.1	(m) Attribute:	Protocol Identity
13.12.3	(c) Constraint:	Organizationally Unique Identifier = 00-12-0F
13.12.3.1	(m) Attribute:	List of IEEE 802.3 Subtypes
13.12.3.1.1	(m) Attribute:	Subtype
13.12.3.1.2	(c) Constraint:	Subtype = MAC_PHY_CONFIGURATION_STATUS
13.12.3.1.2.1	(m) Attribute:	Auto Negotiation Support And Status
13.12.3.1.2.2	(m) Attribute:	Advertised Capability
13.12.3.1.2.3	(m) Attribute:	Operational MAU type
13.12.3.1.3	(c) Constraint:	Subtype = POWER_VIA_MDI
13.12.3.1.3.1	(m) Attribute:	MDI Power Support
13.12.3.1.3.2	(m) Attribute:	Power Pair
13.12.3.1.3.3	(m) Attribute:	Power Class
13.12.3.1.4	(c) Constraint:	Subtype = LINK_AGGREGATION
13.12.3.1.4.1	(m) Attribute:	Link Aggregation Status
13.12.3.1.4.2	(m) Attribute:	Aggregated Port ID
13.12.3.1.5	(c) Constraint:	Subtype = MAXIMUM_FRAME_SIZE
13.12.3.1.5.1	(m) Attribute:	Maximum Frame Size
13.12.4	(c) Constraint:	Organizationally Unique Identifier = 00-0E-CF
13.12.4.1	(m) Attribute:	List of PNIO Subtypes
13.12.4.1.1	(m) Attribute:	Subtype
13.12.4.1.2	(c) Constraint:	Subtype = LLDP_PNIO_DELAY
13.12.4.1.2.1	(m) Attribute:	PnioDelay
13.12.4.1.2.1.1	(m) Attribute:	PortRxDelayLocal
13.12.4.1.2.1.2	(m) Attribute:	PortRxDelayRemote
13.12.4.1.2.1.3	(m) Attribute:	PortTxDelayLocal
13.12.4.1.2.1.4	(m) Attribute:	PortTxDelayRemote
13.12.4.1.2.1.5	(m) Attribute:	CableDelayLocal
13.12.4.1.3	(c) Constraint:	Subtype = LLDP_PNIO_PORTSTATUS
13.12.4.1.3.1	(m) Attribute:	PnioPortStatus

13.12.4.1.3.1.1	(m) Attribute:	RTClass2PortStatus
13.12.4.1.3.1.2	(m) Attribute:	RTClass3PortStatus
13.12.4.1.3.1.2.1	(m) Attribute:	Status
13.12.4.1.3.1.2.2	(m) Attribute:	Mode
13.12.4.1.4	(c) Constraint:	Subtype = LLDP_PNIO_ALIAS
13.12.4.1.4.1	(m) Attribute:	PnioAlias
13.12.4.1.5	(c) Constraint:	Subtype = LLDP_PNIO_MRP_PORT_STATUS
13.12.4.1.5.1	(m) Attribute:	PnioMRPPortStatus
13.12.4.1.5.1.1	(m) Attribute:	MRPDomainUUID
13.12.4.1.6	(c) Constraint:	Subtype = LLDP_PNIO_INTERFACE_MAC_ADDRESS
13.12.4.1.6.1	(m) Attribute:	PnioInterfaceMacAddress
13.12.4.1.7	(c) Constraint:	Subtype = LLDP_PNIO_PTCP_STATUS
13.12.4.1.7.1	(m) Attribute:	PnioPTCPStatus
13.12.4.1.7.1.1	(m) Attribute:	PnioPTCPMasterSourceAddress
13.12.4.1.7.1.2	(m) Attribute:	PnioPTCPSubdomainUUID
13.12.4.1.7.1.3	(m) Attribute:	PnioIRDataUUID
13.12.4.1.7.1.4	(m) Attribute:	PnioLengthOfPeriod
13.12.4.1.7.1.4.1	(m) Attribute:	Length
13.12.4.1.7.1.4.2	(m) Attribute:	Validity
13.12.4.1.7.1.5	(m) Attribute:	PnioRedPeriodBegin
13.12.4.1.7.1.6	(m) Attribute:	PnioOrangePeriodBegin
13.12.4.1.7.1.7	(m) Attribute:	PnioGreenPeriodBegin

SERVICES:

1	(m) OpsService:	Remote Systems Data Change

6.3.7.2.2 Attributes

Implicit

The key attribute Implicit indicates that the object is implicitly addressed by the service.

Multiple Interface Mode

This attribute controls the behavior of the IEEE 802.1AB ASE regarding the location of the name of port and the NameOfStation.

The value FALSE indicates that the attribute Chassis ID contains the NameOfStation and the attribute PortID contains the name of the port.

The value TRUE indicates that the attribute Chassis ID is defined by local means and the attribute Port ID contains the name of the port + "." + the NameOfStation.

Attribute type: Boolean

Default value: FALSE

Chassis ID

This non-volatile attribute contains the Chassis ID which shall be used as Chassis ID subtype locally assigned compliant with IEEE 802.1AB. This field shall contain a VisibleString indicating the specific identifier for the particular chassis in this system. The value of the attribute is dependent on the attribute Multiple Interface Mode defined below. If Multiple Interface Mode is set to FALSE this attribute contains the NameOfStation, otherwise the content of this attribute is defined by local means.

Attribute type: VisibleString[240]

Port ID

This attribute contains the name of the local port which shall be used as Port ID subtype locally assigned compliant with IEEE 802.1AB.

If the attribute Multiple Interface Mode is set to FALSE, the attribute contains the name of the port in the form "port-xyz" or "port-xyz-rstuv" where x, y, z is in the range "0"-"9" from 001 up

to 255 and r, s, t, u, v is in the range “0”-“9” from 00 000 up to 65 535. The values “port-001” or “port-001-00000” shall be used for the first port submodule for an interface within slot 0. The values x, y, z shall be used to identify the port number. The values r, s, t, u, v shall be used to identify the slot.

If the attribute Multiple Interface Mode is set to TRUE, the attribute Own Port ID contains the name of the port (“port-xyz” or “port-xyz-rstuv”) + “.” + the NameOfStation.

Attribute type: VisibleString[240]

NOTE 1 Within the context of LLDP protocol a DTE is referred to as port. It does not mean a port from the UDP context.

LLDP Time To Live

This attribute indicates the number of seconds that the recipient LLDP agent is to regard the information associated with this chassis and port identifier to be valid. It shall be compliant with IEEE 802.1AB.

Attribute type: Unsigned16

Allowed Values: 20

Port Description

This attribute contains an alpha-numeric string that indicates the port description. It shall be compliant with IEEE 802.1AB.

If IETF RFC 2863 is implemented, the use of the ifDescr object is recommended for this field.

System Name

This attribute contains an alpha-numeric string that indicates the system’s administratively-assigned name. It shall be compliant with IEEE 802.1AB.

This should be the system’s fully-qualified domain name. If implementations support IETF RFC 3418, the use of the sysName object is recommended for this field.

System Description

This attribute contains an alpha-numeric string that is the textual description of the network entity. It shall be compliant with IEEE 802.1AB.

This value should include the full name and version identification of the system’s hardware type, software operating system, and networking software. If implementations support IETF RFC 3418, the use of the sysDescr object is recommended for this field.

System Capabilities

This attribute contains a bit-map of the capabilities that define the primary function(s) of the system. The bit positions for each function and the associated MIB or standard that are likely, but not guaranteed, to be supported are listed in Table 78. A binary one in the associated bit indicates the existence of that capability. It shall be compliant with IEEE 802.1AB.

NOTE 2 Individual systems may indicate more than one implemented functional capability (for example, both a bridge and router capability).

Table 78 – System capabilities

Position	Meaning
Bit 0	Other
Bit 1	Repeater
Bit 2	Bridge
Bit 3	WLAN Access Point
Bit 4	Router

Position	Meaning
Bit 5	Telephone
Bit 6	DOC SIS cable device
Bit 7	Station Only

Management Address

This attribute contains an OctetString indicating the particular management address. It shall be compliant with IEEE 802.1AB.

The returned address should be the most appropriate for management use, typically a layer-3 address such as the IPv4 address, '192.168.254.10'.

Object Identifier

This attribute identifies the type of hardware component or protocol entity associated with the indicated management address. If no object identifier is available, this field shall not be provided. It shall be compliant with IEEE 802.1AB.

NOTE 3 The interface number and object identifier are included to assist NMS discovery by indicating Enterprise-specific or other starting points for the search, such as the Interface (see IETF RFC 2863) or Entity MIB (see IETF RFC 2737).

List of Organizationally-specific extensions

This attribute list contains the following attributes:

Organizationally Unique Identifier

This attribute contains an OUI that is provided to allow different organizations, such as IEEE 802, IETF, as well as individual software and equipment vendors, to define Type Length Values (TLV) that advertise information to remote entities.

List of IEEE 802.1 Subtypes

This attribute list contains the following attributes:

Subtype

This attribute shall be compliant with IEEE 802.1AB.

Allowed values: PORT_VLAN_ID, PORT_AND_PROTOCOL_VLAN_ID, VLAN_NAME, PROTOCOL_IDENTITY

Port VLAN Identifier

This attribute contains the VLAN ID for the bridge port as defined in IEEE 802.1Q. A value of zero shall be used if the system either does not know the port VLAN identifier or does not support port-based VLAN operation.

Flags

This attribute contains a bit map indicating the port and protocol VLAN capability and status.

Allowed Values: PPVID_SUPPORTED, PPVID_ENABLED

PPVID Reference Number

This attribute contains the port and protocol VALN ID reference number for this 802 LAN station.

If the port is not capable of supporting port and protocol VLANs and/or the port is not enabled with any port protocol VLAN, the PPVID number should be zero.

VLAN ID

This attribute contains the VID number associated with the VLAN name.

VLAN Name

This attribute contains the VLAN's name.

Attribute type: OctetString[32]

If implementations support IETF RFC 2674, the use of the dot1QVLANStaticName object is recommended for this field.

List of Protocol Identities

This attribute list contains the following attributes:

Protocol Identity

This attribute contains the first *n* octets of the protocol after the layer-2 addresses (i.e., starting with the length/type field) that the sender would like to advertise. The value of *n* is determined by the need for the protocol to disambiguate itself. The protocol information string shall include enough octets to allow the receiver to correctly identify the protocol and its version.

NOTE 4 Spanning tree protocols, for example, would need to include five octets: LLC address (two octets), Protocol ID (two octets), plus the protocol version (one octet).

List of IEEE 802.3 Subtypes

This attribute list contains the following attributes:

Subtype

This attribute shall be compliant with IEEE 802.1AB.

Allowed values: MAC_PHY_CONFIGURATION_STATUS, POWER_VIA_MDI, LINK_AGGREGATION, MAXIMUM_FRAME_SIZE

Auto Negotiation Support And Status

This attribute contains a bit map that identifies the auto-negotiation support and current status of the local IEEE 802.3 LAN station as defined in Table 79. If auto-negotiation support bit (bit-0) is one and the auto-negotiation status bit (bit-1) is zero, the IEEE 802.3 physical media dependent sublayer operating mode will be determined the default configuration field value rather than by auto-negotiation.

Table 79 – Auto negotiation support and status

Position	Meaning
Bit 0	Autonegotiation Support (0 – not supported, 1 – supported)
Bit 1	Autonegotiation Status (0 – disabled, 1 – enabled)

NOTE 5 This attribute can be written via Write PD Port Data Adjust.

Advertised Capability

This attribute contains an integer value as defined by the ifMauAutoNegCapAdvertisedBits object in IETF RFC 4836.

Operational MAU type

This attribute is an integer value indicating the operational MAU type of the sending device. This value is derived from the list position of the corresponding dot3MauType as listed in IETF RFC 4836 (or subsequent revisions) and is equal to the last number in the respective dot3MauType. For MAU types not listed in IETF RFC 4836 (or subsequent revisions), the value of this field shall be set to zero.

MDI Power Support

This attribute contains a bit-map of the MDI power capabilities and status as defined in Table 80.

Table 80 – MDI Power Support

Position	Meaning

Position	Meaning
Bit 0	Port Class (0 – Power Sourcing Equipment, 1 – Powered Device)
Bit 1	MDI Power Support (0 – not supported, 1 – supported)
Bit 2	MDI Power State (0 – disabled, 1 – enabled)
Bit 3	Pairs Control Ability (0 - pair selection cannot be controlled, 1 - pair selection can be controlled)

Power Pair

This attribute contains an integer value as defined by the pethPsePortPowerPairs object in IETF RFC 3621.

Power Class

This attribute contains an integer value as defined by the pethPsePortPowerClassification object in IETF RFC 3621.

Link Aggregation Status

This attribute contains a bit map of the link aggregation capabilities and the current aggregation status of the link as defined in Table 81.

Table 81 – Link aggregation status

Position	Meaning
Bit 0	Aggregation Capability (0 – not supported, 1 – supported)
Bit 1	Aggregation Status (0 – disabled, 1 – enabled)

Aggregated Port ID

This attribute contains the ISO/IEC 15802-1 aggregated port identifier. An AggPortID is derived from the ifNumber in the ifIndex for the interface.

Maximum Frame Size

This attribute contains an integer value indicating the maximum supported frame size in octets shall be set to 1 522.

List of PNIO Subtypes

This attribute list contains the following attributes:

Subtype

This attribute contains the following values.

Allowed values: LLDP_PNIO_DELAY, LLDP_PNIO_PORTSTATUS, LLDP_PNIO_ALIAS, LLDP_PNIO_MRP_PORT_STATUS, LLDP_PNIO_INTERFACE_MAC_ADDRESS, LLDP_PNIO_PTCP_STATUS

PnioDelay

This attribute contain of the following attributes:

PortRxDelayLocal

This attribute contains the local receiving delay of the port in nanoseconds.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFF

PortRxDelayRemote

This attribute contains the local receiving delay of the peer port in nanoseconds.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFF

PortTxDelayLocal

This attribute contains the local transmission delay of the port in nanoseconds.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFF

PortTxDelayRemote

This attribute contains the local transmission delay of the peer port in nanoseconds.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFF

CableDelayLocal

This attribute contains the local cable delay in nanoseconds.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFFF

PnioPortStatus

This attribute contain of the following attributes:

RTClass3PortStatus

This attribute contains the following attributes:

Status

This attribute contains the port status for RT_CLASS_3.

Allowed values: NOT_USED, IRDATA_CONFIGURED, RTACCLASS3_NEXT_ACTIVE, RTACCLASS3_NEXT_LOADED, RED_ACTIVE

Mode

This attribute contains the port mode for RT_CLASS_3.

Allowed values: STANDARD, OPTIMISED

PnioAlias

This attribute contains the following VisibleString:

"Value of attribute Port ID" + "." + "Value of Chassis ID"

Attribute type: VisibleString

PnioMRPPortStatus

This attribute shall consist of the following attributes.

MRPDomainUUID

This attribute contains the MRP domain UUID.

Attribute type: UUID

PnioInterfaceMacAddress

This attribute contains the IEEE 802 MAC address.

Attribute type: OctetString[6]

PnioPTCPStatus

This attribute shall consist of the following attributes.

PnioPTCPMasterSourceAddress

This attribute contains the IEEE 802 MAC address of the PTCP master.

Attribute type: OctetString[6]

PnioPTCPSubdomainUUID

This attribute contains the UUID of the PTCP subdomain.

Attribute type: UUID

PnioIRDataUUID

This attribute contains the UUID of the Isochronous Relay (IR) data.

Attribute type: UUID

PnioLengthOfPeriod

This attribute shall consist of the following attributes.

Length

This attribute contains the value for the duration of the cycle in nanoseconds. The value shall be a multiply of 31 250 nanoseconds.

Attribute type: Unsigned32

Allowed values: 0x00007A12 to 0x003D0900 if valid, 0 if invalid

Validity

This attribute contains the validity of the length.

Allowed values: INVALID, VALID

PnioRedPeriodBegin

This attribute shall consist of the following attributes.

Offset

This attribute contains the value for the offset related to the begin of the cycle in nanoseconds.

Attribute type: Unsigned32

Allowed values: 0x00000000 to 0x003D08FF if valid, 0 if invalid

Validity

This attribute contains the validity of the offset.

Allowed values: INVALID, VALID

PnioOrangePeriodBegin

This attribute shall consist of the following attributes.

Offset

This attribute contains the value for the offset related to the begin of the cycle in nanoseconds.

Attribute type: Unsigned32

Allowed values: 0x00000000 to 0x003D08FF if valid, 0 if invalid; this standard requires value 0.

Validity

This attribute contains the validity of the offset.

Allowed values: INVALID, VALID; this standard requires value VALID.

PnioGreenPeriodBegin

This attribute shall consist of the following attributes.

Offset

This attribute contains the value for the offset related to the begin of the cycle in nanoseconds.

Attribute type: Unsigned32

Allowed values: 0x00000000 to 0x003D08FF if valid, 0 if invalid

Validity

This attribute contains the validity of the offset.

Allowed values: INVALID, VALID

List of Remote Systems Data

This attribute list contains the same attributes as specified for Real List of Ports including all subsequent attributes until Maximum Frame Size. Furthermore, the following additional attributes are included.

Port ID

This attribute has the same definition as Own Port ID.

Chassis ID

This attribute has the same definition as Interface Name.

MAC Address

This attribute contains the devices physical address according to IEEE 802.3 MAC address.

Attribute type: OctetString[6]

Line Delay

This attribute contains the measured line delay. The time base shall be 1 ns.

Attribute type: Unsigned32

6.3.7.2.3 Invocation of the IEEE 802.1AB object

For the invocation of the IEEE 802.1AB object the following rules apply:

- IEEE 802.1AB objects shall exist for each interface.

6.3.7.3 IEEE 802.1AB service specification

The protocol according to IEEE 802.1AB shall be used to send and receive the attribute values of the ASE. IEEE 802.1AB uses the services of the LPM to transmit and receive information to and from other LLDP agents.

6.3.7.3.1 Remote systems data change

This local service shall be used to indicate a change of certain Remote Systems Data. Table 82 shows the parameters of the service.

Table 82 – Remote systems data change

Parameter name	ind
Argument	M
Local Port ID	M
List of Neighbors	U

Parameter name	ind
Remote Chassis ID	U
Remote Port ID	U
Remote Cable Delay	U
Remote RT_CLASS_3 Port Status	U
Remote PTCP Status	U
Remote Line Delay	U
Auto Negotiation Support And Status	U
Advertised Capability	U
Operational MAU type	U

Argument

The argument shall convey the service specific parameters of the service indication.

Local Port ID

This parameter is the local identifier.

List of Neighbors

This parameter list consists of the following parameter:

Remote Chassis ID

This parameter contains the value of the attribute of the ASE.

Remote Port ID

This parameter contains the value of the attribute of the ASE.

Remote Cable Delay

This parameter contains the value of the attribute of the ASE.

Remote RT_CLASS_3 Port Status

This parameter contains the value of the attribute of the ASE.

Remote PTCP Status

This parameter contains the value of the attribute of the ASE.

Remote Line Delay

This parameter contains the value of the sum of the following attribute of the ASE:

CableDelayLocal

PortRxDelayLocal

PortTxDelayRemote

Auto Negotiation Support And Status

This parameter contains the value of the attribute of the ASE.

Advertised Capability

This parameter contains the value of the attribute of the ASE.

Operational MAU type

This parameter contains the value of the attribute of the ASE.

6.3.7.3.2 Placeholder

This is an empty subclause to be in line with IEC directive part 2 and the structure of service specifications.

6.3.8 IEEE 802.1AS ASE

6.3.8.1 Overview

The concepts according to IEEE 802.1AS standard shall be applied. This part of the specification defines the utilization of the IEEE 802.1AS standard.

The formal model of the IEEE 802.1AS ASE is presented by the IEEE 802.1AS class specification, containing a description of its attributes and services.

6.3.8.2 IEEE 802.1AS class specification

6.3.8.2.1 Template

An IEEE 802.1AS object is described by the following template:

ASE: IEEE 802.1AS ASE

CLASS: IEEE 802.1AS

CLASS ID: not used

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Key Attribute:	not used
2	(m) Attribute:	Role
3	(m) Attribute:	Time
4	(m) Attribute:	List of Ports
4.1	(m) Attribute:	PortID
4.2	(m) Attribute:	Path Delay Boundary

SERVICES:

1	(m) OpsService:	Local Get Time
2	(m) OpsService:	Local Set Time
3	(m) OpsService:	Local Time State Info

6.3.8.2.2 Attributes

This key attribute is not used.

Expected Role

This attribute identifies the expected role of the interface regarding time synchronization.

Attribute type: Unsigned8

Allowed values: OFF, MASTER, SLAVE

Time

This attribute holds the time and the status.

Attribute type: TimeStamp

List of Ports

This attribute consists of the following elements:

PortID

This attribute identifies one Port of a bridge.

Attribute type: Unsigned8

Path Delay Boundary

This attribute shall contain the value ON for end-ports of a 802.1AS Domain.

Attribute type: Unsigned8

Allowed values: ON, OFF

6.3.8.3 IEEE 802.1AS service specification

The protocol according to IEEE 802.1AB shall be used to send and receive the attribute values of the ASE. IEEE 802.1AB uses the services of the LPM to transmit and receive information to and from other LLDP agents.

6.3.8.3.1 Local Get Time

This local service may be used to get the local time. Table 83 shows the parameters of the service.

Table 83 – Local Get Time

Parameter name	req	cnf
Argument	M	
Result(+)		S(=)
Time		M(=)
Result(-)		S(=)

Argument

The argument shall convey the service specific parameters of the service request.

Result(+)

This parameter indicates that the service request succeeded.

Time

This parameter contains the value of the equivalent ASE attribute.

Result(-)

This parameter indicates that the service request failed.

6.3.8.3.2 Local Set Time

This local service may be used to set the local time. Table 84 shows the parameters of the service.

Table 84 – Local Set Time

Parameter name	req	cnf
Argument	M	
Time	M	
Result(+)		S(=)
Result(-)		S(=)

Argument

The argument shall convey the service specific parameters of the service request.

Time

This parameter contains the value for the equivalent ASE attribute.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

6.3.8.3.3 Local time changed event

This local service shall be used to indicate a change of the local time. This event is initiated whenever the time was set to a new value. Table 85 shows the parameters of the service.

Table 85 – Local time changed event

Parameter name	ind
Argument	M
OldTime	M
NewTime	M

Argument

The argument shall convey the service specific parameters of the service indication.

OldTime

This parameter indicates the point of time in the former time scale, when the new time has been set.

Attribute type: TimeStamp

NewTime

This parameter indicates the value of the time, to which the time has been set.

Attribute type: TimeStamp

6.3.8.4 Invocation of the IEEE 802.1AS object

For the invocation of the IEEE 802.1AS object the following rules apply:

- IEEE 802.1AS objects shall exist for each interface.

6.3.9 IEEE 802.1D ASE**6.3.9.1 Overview**

The IEEE 802.1D ASE controls the bridging according to IEEE 802.1D (MAC bridges) extended with the special requirements for RT_CLASS_3.

The concepts according to IEEE 802.1D shall be applied. This clause defines the utilization of the IEEE 802.1D.

The bridging mechanisms of IEEE 802.1D shall be temporary disabled within the RED phase. In this case, the bridging of frames out of frame ID range 3 (see Table 276) shall be done according to the attributes defined by this ASE with the behavior according to the RT_CLASS_3 Relay protocol machine. The values of the attributes define a special forwarding table.

6.3.9.2 IEEE 802.1D class specification**6.3.9.2.1 General**

The IEEE 802.1D ASE defines one IEEE 802.1D object type:

6.3.9.2.2 Template

An IEEE 802.1D object is described by the following template:

ASE: IEEE 802.1D ASE

CLASS: IEEE 802.1D

CLASS ID: not used

PARENT CLASS: top

ATTRIBUTES:

- 1 (m) Key Attribute: Implicit
- 2 (m) Attribute: PD IR Global Data
 - 2.1 (m) Attribute: IR Data UUID
 - 2.2 (m) Attribute: Max Bridge Delay
 - 2.3 (m) Attribute: Number of Ports
 - 2.4 (m) Attribute: List of Port Delays
 - 2.4.1 (m) Attribute: Max Port Tx Delay
 - 2.4.2 (m) Attribute: Max Port Rx Delay
 - 2.4.3 (m) Attribute: Max Line Rx Delay
 - 2.4.4 (m) Attribute: Yellow Time
- 3 (o) Attribute: PD IR Frame Data
 - 3.1 (m) Attribute: Frame Data Properties
 - 3.2 (o) Attribute: List of PD IR Frame Data Elements
 - 3.2.1 (m) Attribute: Frame Send Offset
 - 3.2.2 (m) Attribute: Data Length
 - 3.2.3 (m) Attribute: Reduction Ratio
 - 3.2.4 (m) Attribute: Phase
 - 3.2.5 (m) Attribute: Frame ID
 - 3.2.6 (m) Attribute: Ethertype
 - 3.2.7 (m) Attribute: Rx Port
 - 3.2.8 (m) Attribute: Frame Details
 - 3.2.8.1 (m) Attribute: Sync Frame
 - 3.2.8.2 (m) Attribute: Meaning Frame Send Offset
 - 3.2.9 (m) Attribute: Tx Port Group
 - 3.2.9.1 (m) Attribute: Number Of Tx Port Group Array Elements
 - 3.2.9.2 (m) Attribute: Tx Port Group Array
- 4 (m) Attribute: PD IR Begin End Data
 - 4.1 (m) Attribute: Red Guard
 - 4.1.1 (m) Attribute: Start of Red Frame ID
 - 4.1.2 (m) Attribute: End of Red Frame ID
 - 4.2 (m) Attribute: Number of Ports
 - 4.3 (m) Attribute: List of Ports
 - 4.3.1 (m) Attribute: Number of Assignments
 - 4.3.2 (m) Attribute: List of Assignments
 - 4.3.2.1 (m) Attribute: Tx Begin End Assignment
 - 4.3.2.1.1 (m) Attribute: Red Orange Period Begin
 - 4.3.2.1.2 (m) Attribute: Orange Period Begin
 - 4.3.2.1.3 (m) Attribute: Green Period Begin
 - 4.3.2.2 (m) Attribute: Rx Begin End Assignment
 - 4.3.2.2.1 (m) Attribute: Red Orange Period Begin
 - 4.3.2.2.2 (m) Attribute: Orange Period Begin
 - 4.3.2.2.3 (m) Attribute: Green Period Begin
 - 4.3.3 (m) Attribute: Number of Phases
 - 4.3.4 (m) Attribute: List of Phases
 - 4.3.4.1 (m) Attribute: Tx Phase Assignment

4.3.4.1.1	(m) Attribute:	Assigned Value For Reserved Begin
4.3.4.1.2	(m) Attribute:	Assigned Value For Orange Begin
4.3.4.1.3	(m) Attribute:	Assigned Value For Reserved End
4.3.4.2	(m) Attribute:	Rx Phase Assignment
4.3.4.2.1	(m) Attribute:	Assigned Value For Reserved Begin
4.3.4.2.2	(m) Attribute:	Assigned Value For Orange Begin
4.3.4.2.3	(m) Attribute:	Assigned Value For Reserved End
5	(m) Attribute:	List of Ports
5.1	(m) Attribute:	Port
5.1.1	(m) Attribute:	Transmission Period Sending
5.1.2	(m) Attribute:	Transmission Period Receiving
5.1.3	(m) Attribute:	Local Port State
5.1.4	(m) Attribute:	Local RTClass 3 Port State
6	(o) Attribute:	Clock Domain Phase Counter

SERVICES:

1	(m) OpsService:	Port State Change
2	(o) OpsService:	Set Port State
3	(o) OpsService:	Flush filtering data base

6.3.9.2.3 Attributes**Implicit**

The key attribute Implicit indicates that the IEEE 802.1D object is implicitly addressed by the service.

PD IR Global Data

This attribute contains the global IR data used by the engineering system as computation base. The device shall adjust its physical characteristics to match the global data. The attribute is composed of the following elements:

IR Data UUID

This attribute contains the unique identifier of the isochronous realtime project which is configured by the engineering system.

Attribute type: UUID

Max Bridge Delay

This attribute indicates the maximum bridge delay of the interface in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000 ("unknown"), 0x00000001 to 0x3B9AC9FF

Number of Ports

This attribute indicates the number of ports the List of Port Delays contains.

Attribute Type: Unsigned32

Allowed Values: 0x00000001 to 0x000000FF

List of Port Delays

This list contains the port transmit and receive delays for each port, starting with Port 1 up to the port number indicated by the attribute Number of Ports. A list element is composed of the following elements:

Max Port Tx Delay

This attribute indicates the maximum port transmission delay in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000 (“unknown”), 0x00000001 to 0x3B9AC9FF

Max Port Rx Delay

This attribute indicates the maximum port receive delay in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000 (“unknown”), 0x00000001 to 0x3B9AC9FF

Max Line Rx Delay

This attribute indicates the maximum line receive delay in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000 (“no delay, forward as soon as possible”), 0x00000001 to 0x3B9AC9FF

Yellow Time

This attribute indicates the yellow time in multiples of 1 ns.

The yellow time defines the length of the YELLOW period. It allows a bridge to optimize the forwarding of frames belonging to the protocols PTCP, MRP, RT and IEEE 802.1AS.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00001AE0 to 0x0001E848

PD IR Frame Data

This attribute contains the global IR data used by the engineering system as computation base. The device shall adjust its physical characteristics to match the global data. The attribute is composed of the following elements:

Frame Data Properties

This attribute contains of the following elements:

Forwarding Mode

This attribute describes the settings for the forwarding mode being used for RT_CLASS_3 frames.

Attribute Type: Unsigned8

The allowed values are shown in Table 86:

Table 86 – Allowed values of Forwarding Mode

Value(decimal)	Meaning
0	Absolute mode The FrameSendOffset contains the sending time for a forwarded frame
1	Relative mode The FrameSendOffset shall contain the sending time of a forwarded frame or the “Best effort” value.

Fast Forwarding Multicast MAC Add

This attribute describes the settings for the destination address being used for RT_CLASS_3 frames.

Attribute Type: Unsigned8

The allowed values are shown in Table 87:

Table 87 – Allowed values of Fast Forwarding Multicast MAC Add

Value(decimal)	Meaning
0	Use interface MAC destination unicast address (legacy)
1	Use RT_CLASS_3 destination multicast address
2	Use Fast Forwarding Multicast MAC address

List of PD IR Frame Data Elements

This list contains for all frames in the frame ID range 3 (see Table 276), which the device shall feed in the path information (i.e. on which ports a particular frame is to be forwarded) and time information (i.e. when a particular frame is to be sent). Additionally, if forwarding mode is set to absolute mode, the list contains path and time information for those frames in the frame ID range 3, which shall be forward.

The path information is described by the attribute Tx Port Group, the time information by Frame Send Offset.

Tx Port Group describes, on which ports a particular frame is to be scheduled.

Typically the Frame Send Offset is set to a value composed by the engineering system. A special case is setting Frame Send Offset to 0, indicating that the frame is to be sent “as fast as possible”; i.e. within RED the frame will leave the device with a slight time variation. For a frame fed in by the device, the Frame Send Offset indicated here has to match the Frame Send Offset as indicated in the CRL class.

If the forwarding mode is set to “relative mode”, the device acts as a relative forwarder; however since besides the time information also the path information is missing for frames to be forward, this use case is only applicable for a two port device. The rules for frames to forward are as following

If a particular singlecast RT_CLASS_3 frame is received on some port and the device is not end point for it, the frame is transmitted with a constant delay on the other port.

If a particular multicast RT_CLASS_3 frame is received on some port, the frame is transmitted with a constant delay on the other port; independent whether the device is end point for it or not.

The list is composed of the following list elements:

Frame Send Offset

This attribute indicates the offset relative to the beginning of the related send clock cycle when the particular frame shall be sent in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000 to 0x003D08FF

NOTE 1 The minimal planned gap between two frames out of frame ID range 3 (see Table 276) is 1 120 ns.

Data Length

This attribute contains the overall length of the transmitted Data (CSDU or else CSFSDU Length); thus for a dynamically packed frame it indicates the length of all sub frames including the sub frame this IO device contributes.

Attribute type: Unsigned16

Allowed Values for RT_CLASS_3: 40 to 1 440

NOTE 2 The lower bound of the C_SDU length is derived from the minimum frame size of an Ethernet frame (= 64 bytes). On subtracting the size of the header (i.e. destination address, source address and type) and the size of the footer (i.e. CRC) 46 bytes are left. On subtracting the size of the RT frame header (i.e. Frame ID) and footer (i.e. cycle counter, data status and transfer status) the minimum length of the C_SDU is 40 bytes.

Reduction Ratio

This attribute contains the reduction ratio of the frame compared to the send clock.

Attribute type: Unsigned16

The allowed values are shown in Table 88:

Table 88 – Allowed values of Reduction Ratio

Value(decimal)	Meaning
1	Mandatory
2	Mandatory
4	Mandatory
8	Mandatory
16	Mandatory
32	Optional
64	Optional
128	Optional
256	Optional
512	Optional
1 024 – 65 535	Reserved
other	Optional

Phase

This attribute indicates the phase when the particular frame shall be sent.

Attribute Type: Unsigned16

Allowed Values: 1 until current value of the attribute Reduction Ratio

Frame ID

This attribute contains the identifier of the particular frame which is configured by the engineering system.

Attribute type: Unsigned32

The allowed values according to Table 89 shall be used.

Table 89 – Frame ID

Value (hexadecimal)	Meaning	Use
0x0100 – 0x06FF	Dedicated to RT_CLASS_3 (RED) unicast and multicast	RED, non redundant, normal or DFP

Value (hexadecimal)	Meaning	Use
0x0700 – 0x0FFF	Dedicated to RT_CLASS_3 (RED) unicast and multicast	RED, redundant, normal or DFP
Other	Reserved	Shall not be used for RT_CLASS_3

Ethertype

This attribute indicates the Ethertype to use for the frame.

Attribute Type: Unsigned16

Allowed Values: 0x8892

Rx Port

This attribute contains the number of the receiving port. The value LOCAL_INJECTION shall be used for frames fed in by this device. The value PORT_1 to PORT_255 shall be used for frames to be forwarded.

Attribute type: Unsigned8

Allowed Values: LOCAL_INJECTION, PORT_1 to PORT_255

Frame Details

This attribute consists of the following elements:

Sync Frame

This attributes indicates whether the particular frame is a sync frame or a normal frame.

Attribute Type: Unsigned8

Allowed Values: NO_SYNC_FRAME, PRIMARY_SYNC_FRAME, SECONDARY_SYNC_FRAME

Meaning Frame Send Offset

This attributes indicates the detailed meaning of the attribute Frame Send Offset.

The attribute shall be set to TX_FRAME, when the attribute Frame Send Offset indicates the time, when a frame out of frame ID range 3 (see Table 276) is to be transmitted.

NOTE 3 A frame is locally received, when the attribute Rx Port is set to LOCAL_INJECTION.

Attribute Type: Unsigned8

Allowed Values: TX_FRAME

Tx Port Group

This attribute indicates where the particular frame shall be transmitted. It consists of the following elements:

Number Of Tx Port Group Array Elements

This field shall count the succeeding Tx Port Group Array entries.

Attribute type: Unsigned8

Allowed values: 1, 3, 5, 7, 9, 11, 13, 15, ..., 31, 33.

Tx Port Group Array

The Tx Port Group Array is an array of octets that shall contain at least one and at most 33 octets referred to as Tx Port Group Array entry. A Tx Port Group Array Element shall consist of at least one and at most 8 TxPort entries referred to as TxPortGroup entry 0 to TxPortGroup entry 7. Therefore, the number of

TxPortGroup octet corresponds to the number of ports within a device and shall be calculated as follows:

$$N = M_{\text{highest}} \text{ DIV } 8 + 1 \quad (20)$$

where

N is the number of TxPortGroup octets or the number of array elements,

M_{highest} is the highest number of TxPorts within a device (maximum 255).

The unused TxPort entries shall be set to zero.

The last TxPortGroup octet (octet N) may not contain all 8 TxPort entries. If $M_{\text{highest}} \text{ MOD } 8 \neq 7$ the octet N is not fully filled and the remaining bits shall be set to zero referred to as Padding Bits.

NOTE 4 The term DIV stands for division without remainder. The term MOD stands for the rest of the division.

Bit 0: TxPortEntry_0

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \text{ MOD } 8 = 0$ is present. A padding bit shall be used if no TxPort is present.

Table 90 – Tx Port Entry

Value (hexadecimal)	Meaning
0x00	Transmission off
0x01	Transmission on

The TxPort of local injection is always placed in TxPortEntry_1 of the TxPortGroup octet number one. Thus local injection is mapped onto port 0.

Bit 1: TxPortEntry_1

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \text{ MOD } 8 = 1$ is present. A padding bit shall be used if no TxPort is present.

Bit 2: TxPortEntry_2

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \text{ MOD } 8 = 2$ is present. A padding bit shall be used if no TxPort is present.

Bit 3: TxPortEntry_3

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \text{ MOD } 8 = 3$ is present. A padding bit shall be used if no TxPort is present.

Bit 4: TxPortEntry_4

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \text{ MOD } 8 = 4$ is present. A padding bit shall be used if no TxPort is present.

Bit 5: TxPortEntry_5

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \text{ MOD } 8 = 5$ is present. A padding bit shall be used if no TxPort is present.

Bit 6: TxPortEntry_6

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \bmod 8 = 6$ is present. A padding bit shall be used if no TxPort is present.

Bit 7: TxPortEntry_7

This bit shall be set with the values according to Table 90 if the TxPort with the number that meets $m \bmod 8 = 7$ is present. A padding bit shall be used if no TxPort is present.

Where m is the number of the current TxPort with $1 \leq m \leq N$.

PD IR Begin End Data List

This list contains the Red Guard and the bandwidth administration information for the traffic classes RED, ORANGE and GREEN. It is composed of the following list elements:

Red Guard

This attribute controls the behavior of the receiving ports on Realtime Class 3 Port State Machine (RTC3PSM) being in state "Run", i.e. the Rx port is in mode RED.

The Red Guard prevents wrong frames of entering the traffic class RED as a first filtering layer. All frames with a frame ID below Start of Red Frame ID or above End of Red Frame ID are blocked during the RED period.

The second filtering layer is provided by the IR Frame Data Elements. Only a frame crossing both filters is entering the traffic class RED. If the IR Frame Data is missing, the second filtering layer is omitted.

Start of Red Frame ID

This attribute defines the lower boundary of the Red Guard. The value shall be set to a lower value than the content of the attribute End of Red Frame ID.

Attribute type: Unsigned16

Allowed values: 0x0080 to 0x7FFF

Default value: 0x0100

End of Red Frame ID

This attribute defines the upper boundary of the Red Guard.

Attribute type: Unsigned16

Allowed values: 0x0080 to 0x7FFF

Default value: 0x7FFF

Number of Ports

This attribute contains the number of elements in the List of Ports.

Attribute Type: Unsigned32

Allowed Values: 0x00000001 to 0x000000FF

List of Ports

This list contains the bandwidth administration information. It is composed of the following list elements.

Number of Assignments

This attribute contains the number of elements in the List of Assignments. A device has to support at a minimum 5 different assignments (four for phases with reserved periods (RED) and one for phases without any reserved period).

Within the assignments at a minimum up to two different Red Orange Period Begin and Orange Period Begin values (one of them for phases without any reserved period) and up to five different Green Period Begin values (one of them for phases without any reserved period) shall be supported.

A phase without any reserved period is indicated by having Red Orange Period Begin, Orange Period Begin and Green Period Begin set to 0x0.

The number of assignments supported by the device is indicated by GSD.

Attribute Type: Unsigned32

Allowed Values: 0x00000005 to 0x00000010

List of Assignments

This list contains the assignments. A particular list element contains bandwidth information. A list element is composed of the following attributes:

Tx Begin End Assignment

This attribute defines begins of the RED, ORANGE and GREEN period.

Red Orange Period Begin

This attribute indicates the offset relative to the beginning of the send clock cycle when the RED period shall begin in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000

Orange Period Begin

This attribute indicates the offset relative to the beginning of the send clock cycle when the ORANGE period shall begin in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: shall be set to Green Period Begin (meaning no ORANGE period is defined)

NOTE 5 An orange period exists only for legacy devices.

Green Period Begin

This attribute indicates the offset relative to the beginning of the send clock cycle when the GREEN period shall begin in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000 to 0x003D08FF

Rx Begin End Assignment

This attribute defines begins of the RED, ORANGE and GREEN period.

Red Orange Period Begin

This attribute indicates the offset relative to the beginning of the send clock cycle when the RED period shall begin in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: 0x00000000

Orange Period Begin

This attribute indicates the offset relative to the beginning of the send clock cycle when the ORANGE period shall begin in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: shall be set to Green Period Begin

NOTE 6 For a Rx port there exists no orange period.

Green Period Begin

This attribute indicates the offset relative to the beginning of the send clock cycle when the GREEN period shall begin in multiples of 1 ns.

Attribute Type: Unsigned32

Time base: 1 ns

Allowed Values: shall be set to the end of the RED period.

Number of Phases

This attribute contains the number of elements in the List of Phases used for assignments. Up to 16 different phase assignments are supported.

A particular frame's phase is mapped onto the assignment phase by the Formula (21)

$$\text{Assignment Phase} = \text{Frame Phase} \text{ MOD } \text{Number of Phases}. \quad (21)$$

where

Assignment Phase is the assignment phase
Frame Phase is the particular frame phase
Number of Phases is the number of elements in the List of Phases

Attribute Type: Unsigned32

Allowed Values: 1, 2, 4, 8, 16

List of Phases

This attribute assigns to a particular phase the appropriate period begins.

For the assignment of the particular begins of that phase, the rules as stated in Table 91 apply.

Table 91 – Dependencies of RedOrangePeriodBegin, OrangePeriodBegin, and GreenPeriodBegin

Value (hexadecimal)	Meaning
RedOrangePeriodBegin < OrangePeriodBegin = GreenPeriodBegin	Only a RED period is defined
RedOrangePeriodBegin = OrangePeriodBegin = GreenPeriodBegin = 0	No reserved period is defined
RedOrangePeriodBegin = OrangePeriodBegin < GreenPeriodBegin	Legacy only
RedOrangePeriodBegin < OrangePeriodBegin < GreenPeriodBegin	Not used

Value (hexadecimal)	Meaning
RedOrangePeriodBegin > OrangePeriodBegin	Not used
RedOrangePeriodBegin > GreenPeriodBegin	Not used
OrangePeriodBegin > GreenPeriodBegin	Not used

Tx Phase Assignment

This attribute assigns the values of the RED, ORANGE and GREEN period on all Tx ports for a particular phase.

Assigned Value For Reserved Begin

This attribute contains the positional number of a particular element in the List of Assignments. The content of the attribute Red Orange Period Begin within element's attribute Tx Begin End Assignment is used as begin of the RED period on the Tx port.

Attribute Type: Unsigned8

Allowed Values: 0 to Number of Assignments - 1

Assigned Value For Orange Begin

This attribute contains the positional number of a particular element in the List of Assignments. The content of the attribute Orange Period Begin within element's attribute Tx Begin End Assignment is used as begin of the ORANGE period on the Tx port.

Attribute Type: Unsigned8

Allowed Values: 0 to Number of Assignments - 1

Assigned Value For Reserved End

This attribute contains the positional number of a particular element in the List of Assignments. The content of the attribute Green Period Begin within element's attribute Tx Begin End Assignment is used as begin of the GREEN period on the Tx port.

Attribute Type: Unsigned8

Allowed Values: 0 to Number of Assignments - 1

Rx Phase Assignment

This attribute assigns the values of the RED, ORANGE and GREEN period on all Rx ports for a particular phase.

Assigned Value For Reserved Begin

This attribute contains the positional number of a particular element in the List of Assignments. The content of the attribute Red Orange Period Begin within element's attribute Rx Begin End Assignment is used as begin of the RED period on the Rx port.

Attribute Type: Unsigned8

Allowed Values: 0 to Number of Assignments - 1

Assigned Value For Orange Begin

This attribute contains the positional number of a particular element in the List of Assignments. The content of the attribute Orange Period Begin within element's attribute Rx Begin End Assignment is used as begin of the ORANGE period on the Rx port.

Attribute Type: Unsigned8

Allowed Values: 0 to Number of Assignments - 1

Assigned Value For Reserved End

This attribute contains the positional number of a particular element in the List of Assignments. The content of the attribute Green Period Begin within element's attribute Rx Begin End Assignment is used as begin of the GREEN period on the Rx port.

Attribute Type: Unsigned8

Allowed Values: 0 to Number of Assignments - 1

List of Ports

This attribute shall consist of the following elements:

Port

This attribute consists of the following elements:

Transmission Period Sending

This attribute contains the value of the current transmission period of the port in sending direction. The scheduling of the sending process according IEEE 802.1Q ASE shall be applied.

The following transmitting rules for RED shall be applied:

The local queuing process shall transmit a frame according to Frame Send Offset and Tx Port Group Array if the calculation of Phase and Reduction Ratio fits Phase Number.

The following forwarding rules for RED shall be applied:

The local queuing process shall transmit a prior received frame according to Frame Send Offset, Data Length, Frame ID, and Rx Port. If the calculation of Phase and Reduction Ratio fits Phase Number, then the frame shall be transmitted on all ports stored in Tx Port Group Array.

Within the YELLOW period the forwarding rules of IEEE 802.1D shall be omitted.

The following transmitting rules shall be applied:

The local queuing process shall transmit a frame according if its size is shorter than the remaining YELLOW interval.

The following forwarding rules shall be applied:

The local queuing process shall transmit a prior received frame if its size is shorter than the remaining YELLOW interval.

Within the ORANGE, and GREEN period, the IEEE 802.1D shall be applied.

The transmission period shall be controlled by a local clock. This clock shall be synchronized over the network using GREEN periods. The duration of all periods shall not exceed the time value calculated by means of the attribute Send Clock Factor, which shall be defined by other ASE. The accuracy is determined by conformance classes.

Attribute type: Unsigned8

Allowed Values: RED (optional), ORANGE (legacy only), GREEN, YELLOW (optional)

NOTE 7 As shown in Figure 8, the sending of frames is divided into different periods at the local port. Each phase is defined by $T_{\text{Sendclock}}$ which is between 31,25 μs and 4 ms. The number of frames of each period depends on the scheduled frames for the current phase and may be also zero.

Transmission Period Receiving

This attribute contains the value of the current transmission period of the port in receiving direction.

Within the RED period, only in time RT_CLASS_3 DLPDUs and in time PTCP Sync DLPDUs shall be forwarded according RED Relay rules. All other received DLPDUs shall be handled by the MAC Relay according to the forwarding rules of IEEE 802.1D.

The following receiving rules for RED shall be applied:

The local queuing process shall receive a frame according to Frame Send Offset, Data Length, Frame ID, Rx Port and the calculation of Phase and Reduction Ratio fits Phase Number.

$$\text{Phase} := (\text{Clock Domain Phase Counter MOD Reduction ratio}) + 1 \quad (22)$$

where

<i>Phase</i>	is the phase value
<i>Clock Domain Phase Counter</i>	is the clock domain phase counter
<i>Reduction ratio</i>	is the reduction ratio

The received IEEE 802.3 source MAC addresses shall not be integrated in the forwarding address table to avoid influencing the transmission of all other frames.

The following receiving rules shall be applied:

Within the ORANGE, and GREEN period, the IEEE 802.1D (MAC Relay) shall be applied.

The transmission period shall be controlled by a local clock. This clock shall be synchronized over the network using GREEN periods. The duration of all periods shall not exceed the time value calculated by means of the attribute Send Clock Factor, which shall be defined by other ASE. The accuracy is determined by conformance classes.

Attribute type: Unsigned8

Allowed Values: RED (optional), ORANGE (legacy only), GREEN

Local Port State

This attribute contains the value for the port according IEEE 802.1D bridge management.

Attribute type: Unsigned8

Allowed Values: DISCARDING, BLOCKED, FORWARDING

Local RTClass 3 Port State

This attribute contains the value for the port according to the RED phase. The value OFF means not used or configured for RED phase or the preconditions could not be reached so far. The value RTCLASS3_UP means that the transmission direction is in the RED phase. The value RTCLASS3_RUN means that both directions are in the RED phase.

Attribute type: Unsigned8

Allowed Values: OFF, RTCLASS3_UP, RTCLASS3_RUN

Clock Domain Phase Counter

This attribute contains the count of passed phases. All nodes used with RT_CLASS_3 shall use the same value. It is gained by PTCP clock synchronization.

Attribute type: Unsigned64

6.3.9.3 IEEE 802.1D service specification

6.3.9.3.1 Port state change

This local service shall be used to indicate a change of certain Local Systems Data. Table 92 shows the parameters of the service.

Table 92 – Port state change

Parameter name	ind
Argument	M
Local Port ID	M
Local Port State	M
Local RTClass 3 Port State	M

Argument

The argument shall convey the service specific parameters of the service indication.

Local Port ID

This parameter is the value of the corresponding attribute of the ASE object.

Local Port State

This parameter is the value of the corresponding attribute of the ASE object.

Local RTClass 3 Port State

This parameter is the value of the corresponding attribute of the ASE object.

6.3.9.3.2 Set port state

This local service shall be used to set a certain Local Systems Data. Table 93 shows the parameters of the service.

Table 93 – Set port state

Parameter name	req
Argument	M
Local Port ID	M
Local Port State	U
Local RTClass 3 Port State	U

Argument

The argument shall convey the service specific parameters of the service indication.

Local Port ID

This parameter is the value of the corresponding attribute of the ASE object.

Local Port State

This parameter is the value of the corresponding attribute of the ASE object.

Local RTClass 3 Port State

This parameter is the value of the corresponding attribute of the ASE object.

6.3.9.3.3 Flush filtering data base

This local service shall be used to flush the local filtering data base. Table 94 shows the parameters of the service.

Table 94 – Flush filtering data base

Parameter name	req
Argument	M

Argument

The argument shall convey the service specific parameters of the service indication.

6.3.9.4 Invocation of the IEEE 802.1D object

For the invocation of the IEEE 802.1D object the following rules apply:

- IEEE 802.1D objects shall exist for each interface
- The services with the IFW prefix shall exist for each RT_CLASS_3 interface

6.3.10 IEEE 802.1Q ASE

6.3.10.1 Overview

The concepts according to IEEE 802.1D standard (MAC bridges) shall be applied. This part of the specification defines the utilization of the IEEE 802.1Q standard (Virtual bridged local area networks). It defines extensions for RT_CLASS_3 regarding prioritization of frames.

Application processes contain data that are transmitted in different priorities according IEEE 802.1Q. Furthermore, the usage of these priorities shall be extended in different time intervals according to this specification. The IEEE 802.1Q ASE defines attributes to determine the timing behavior.

The formal model of the IEEE 802.1Q ASE is presented by the IEEE 802.1Q class specification, containing a description of its attributes.

6.3.10.2 IEEE 802.1Q class specification

6.3.10.2.1 Template

An IEEE 802.1Q object is described by the following template:

ASE: IEEE 802.1Q ASE

CLASS: IEEE 802.1Q

CLASS ID: not used

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Key Attribute: not used
2	(o) Attribute: Network Control High Queue
3	(m) Attribute: Network Control Low Queue
4	(m) Attribute: Prio 7 Send Queue
5	(m) Attribute: Prio 6
5.1	(o) Attribute: Red Send Queue
5.2	(m) Attribute: Prio 6 Send Queue
6	(m) Attribute: Prio 5 Send Queue
7	(m) Attribute: Prio 4 Send Queue
8	(m) Attribute: Prio 3 Send Queue
9	(m) Attribute: Prio 2 Send Queue

10	(m) Attribute:	Prio 1 Send Queue
11	(m) Attribute:	Prio 0 Send Queue
12	(m) Attribute:	Transmission Period
13	(o) Attribute:	Clock Domain Phase Counter

6.3.10.2.2 Attributes

This key attribute is not used.

Network Control High Queue

This optional attribute contains the list of DLPDUs for synchronization for those DLPDUs which are not planned within IR Data records or for those received outside the planned time. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Network Control Low Queue

This attribute contains the list of DLPDUs for IEEE 802.1D organizational priority for management protocols like LLDP, PTCP line delay measurement, MRP and others. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 7 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 7. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 6

This attribute consists of the following attributes:

Red Send Queue

This optional attribute contains the list of DLPDUs for RT_CLASS_3 and for DLPDUs for synchronization which were planned within a IR Data record and received in time. The sequence of frames and the exact sending time shall be determined by other ASE attributes. After transmission the DLPDU shall be removed from the queue. With the start of a new phase, the Red Send Queue shall be filled with the DLPDUs for the current phase.

Attribute type: OctetString

NOTE 1 The application layer for decentralized periphery defines a list of IR Frame Data elements to determine the sequence and exact sending time of frames.

Prio 6 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 6 including DLPDUs for RT_CLASS_1 and RT_CLASS_UDP. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 5 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 5. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 4 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 4. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 3 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 3. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 2 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 2. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 1 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 1. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Prio 0 Send Queue

This attribute contains the list of DLPDUs for IEEE 802.1Q priority 0. After transmission the DLPDU shall be removed from the queue.

Attribute type: OctetString

Transmission Period

This attribute contains the value of the current transmission period of the port. It is aimed to control the usage of the bandwidth within the network.

NOTE 2 In theory, each device could use always wired speed for sending frames. Such behavior applied over a certain amount of time would bring the system out operation. Therefore, a fair mechanism to restrict the sending performance of each device is provided.

Within the RED period, only DLPDUs from the Red Send Queue shall be transmitted.

Within the GREEN period, DLPDUs from all queues shall be transmitted according to their priorities.

Within the GREEN period, for the local scheduling the following rules shall be applied:

- The local queuing process shall drop all RT_CLASS_1 frames for the next phase if at the beginning of a new GREEN period the Prio 6 Send Queue is not empty. Each drop shall be signaled to the local application.
- In average, not more than 90 percent of bandwidth shall be used to avoid overload of the network.

Within the YELLOW period, DLPDUs from all queues shall be transmitted according to their priorities if the transmission of the frame ends within this period. If the calculated transmission time for a frame exceeds the period, the first frame with lesser priority can be tested optionally.

The sequence of periods shall be RED, GREEN, and YELLOW. The RED period may be omitted. The YELLOW period shall exist if RED exists and shall be omitted otherwise.

The transmission period shall be controlled by a local clock. This clock shall be synchronized over the network using GREEN periods. The duration of all periods shall not exceed the time value calculated by means of the attribute Send Clock Factor, which shall be defined by other ASE. The accuracy is determined by conformance classes.

Attribute type: Unsigned8

Allowed Values: RED (optional), GREEN, YELLOW (optional)

NOTE 3 As shown in Figure 8, the sending of frames is divided into different periods at the local port. Each phase is defined by $T_{\text{Sendclock}}$ which is between $31,25 \mu\text{s}$ and 4 ms . The number of frames of each period depends on the scheduled frames for the current phase and may be also zero.

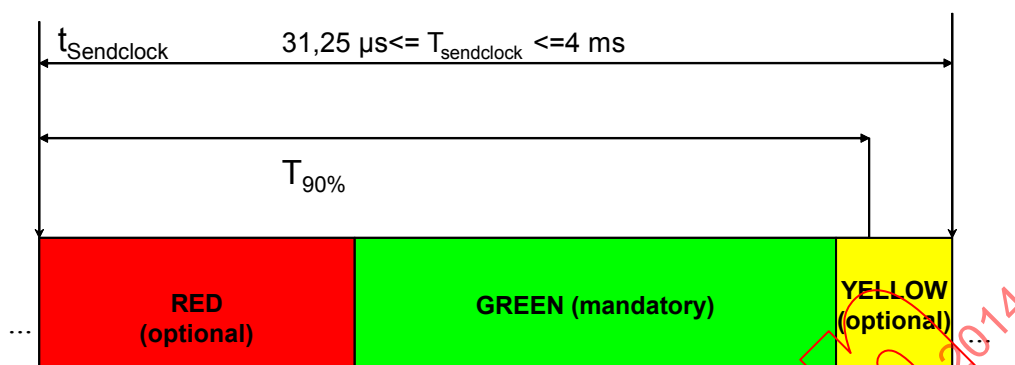


Figure 8 – Example of periods at a local port

The first and the last invocation of the RED period shall be synchronized between the local and the neighborhood port by means of Remote Systems Data Change (Remote RT_CLASS_3 Port Status).

Clock Domain Phase Counter

This attribute contains the count of passed phases. All nodes used with RT_CLASS_3 shall use the same value. It is gained by PTP clock synchronization.

Attribute type: Unsigned64

6.3.10.2.3 Invocation of the IEEE 802.1Q object

For the invocation of the IEEE 802.1Q object the following rules apply:

- IEEE 802.1Q objects shall exist for each port.

6.3.10.3 IEEE 802.1Q service specification

There are no services specified.

6.3.11 IEEE 802.3 ASE

6.3.11.1 Overview

The IEEE 802.3 ASE controls the medium access.

The concepts according to IEEE 802.3 (Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications) shall be applied. This clause defines the utilization of the IEEE 802.3.

6.3.11.2 IEEE 802.3 class specification

6.3.11.2.1 General

The IEEE 802.3 ASE defines one IEEE 802.3 object type.

6.3.11.2.2 Template

An IEEE 802.3 object is described by the following template:

ASE: IEEE 802.3 ASE

CLASS: IEEE 802.3

CLASS ID: not used

PARENT CLASS: top

ATTRIBUTES:

- 1 (m) Key Attribute: Implicit
- 2 (m) Attribute: List of Ports
- 2.1 (m) Attribute: Port
- 2.1.1 (m) Attribute: MAU Type
- 2.1.2 (m) Attribute: Autonegotiation State
- 2.1.3 (m) Attribute: Link State

SERVICES:

- 1 (m) OpsService: M_UNITDATA
- 2 (m) OpsService: MAU Type Change
- 3 (m) OpsService: Set MAU Type

6.3.11.2.3 Attributes

Implicit

The key attribute Implicit indicates that the IEEE 802.3 object is implicitly addressed by the service.

List of Ports

This attribute shall consist of the following elements:

Port

This attribute consists of the following elements:

Attribute type: Unsigned8

MAU Type

This attribute contains the value of the current MAU type according IETF RFC 4836. If the value of the attribute Status is DOWN or OFF, the MAU type shall be zero.

Attribute type: Unsigned16

Autonegotiation State

This attribute contains the value of the current autonegotiation setting. For optical transmission the value shall be set to DISABLE.

Attribute type: Unsigned8

Allowed Values: ENABLE, DISABLE

Link State

This attribute contains the value of the current port status setting. It shall be OFF, if the media attachment unit is disabled. It shall be DOWN, if the media attachment unit is enabled but no Link is detected. It shall be UP, if the media attachment unit is enabled and a Link is detected.

Attribute type: Unsigned8

Allowed Values: UP, DOWN, OFF

6.3.11.3 IEEE 802.3 service specification

6.3.11.3.1 M_UNITDATA

For the definition of this service see IEEE 802.3.

This local service shall be used to transmit (M_UNITDATA.req) or receive (M_UNITDATA.ind) a PDU.

6.3.11.3.2 MAU Type Change

This local service shall be used to indicate a change of certain Local Systems Data. Table 95 shows the parameters of the service.

Table 95 – MAU Type change

Parameter name	ind
Argument	M
Local Port ID	M
Local MAU Type	U
Local Autonegotiation State	U
Local Link State	U

Argument

The argument shall convey the service specific parameters of the service indication.

Local Port ID

This parameter is the value of the corresponding attribute of the ASE object.

Local MAU Type

This parameter is the value of the corresponding attribute of the ASE object.

Local Autonegotiation State

This parameter is the value of the corresponding attribute of the ASE object.

Local Link State

This parameter is the value of the corresponding attribute of the ASE object.

6.3.11.3.3 Set MAU Type

This local service shall be used to indicate a change of certain Remote Systems Data. Table 96 shows the parameters of the service.

Table 96 – Set MAU Type

Parameter name	ind
Argument	M
Local Port ID	M
Local MAU Type	U
Local Autonegotiation State	U
Local Link State	U

Argument

The argument shall convey the service specific parameters of the service indication.

Local Port ID

This parameter is the value of the corresponding attribute of the ASE object.

Local MAU Type

This parameter is the value of the corresponding attribute of the ASE object.

Local Autonegotiation State

This parameter is the value of the corresponding attribute of the ASE object.

Local Link State

This parameter is the value of the corresponding attribute of the ASE object.

6.3.11.4 Invocation of the IEEE 802.3 object

For the invocation of the IEEE 802.3 object the following rules apply:

- IEEE 802.3 objects shall exist for each interface.

6.3.12 Fragmentation ASE

6.3.12.1 Overview

The length of a transmission period is limited on the lower end by the maximum size of an Ethernet frame. The fragmentation reduces, with a peer to peer approach, the conveyed size of an Ethernet frame and allows to use a LengthOfPeriod down to 31,25 µs.

The fragmentation ASE hides the issue fragmentation from the upper layers by acting as an intermediate application service entity and providing the same service interface as the IEEE 802.3 ASE.

For this the Fragmentation ASE is derived from the IEEE 802.3 ASE and overwrites the operational service M_UNITDATA to hide the fragmentation specific details from the upper layers.

Frames to be transmitted which are subject for fragmentation will get fragmented by the operational service and transmitted in fragments by using the basic M_UNITDATA service of the IEEE 802.3 ASE. Fragmented frames are received by the IEEE 802.3 ASE, collected together to restore the original frame and then indicated by the M_UNITDATA indication to the upper layers.

The frames related to the protocols PTCP, MRP and RTC (RT_CLASS_1, RT_CLASS_2 and RT_CLASS_3) are never fragmented.

The fragmentation ASE is optional; it shall be supported, if the residual time for NRT frames drops below 125 µs.

6.3.12.2 Fragmentation class specification

6.3.12.2.1 General

The Fragmentation ASE defines one Fragmentation object type.

6.3.12.2.2 Template

A Fragmentation object is described by the following template:

ASE: Fragmentation ASE

CLASS: Fragmentation

CLASS ID: not used

PARENT CLASS: IEEE 802.3 ASE

ATTRIBUTES:

1 (m) Key Attribute: Implicit

SERVICES:

1 (m) OpsService: M_UNITDATA

6.3.12.2.3 Attributes

Implicit

The key attribute Implicit indicates that the Fragmentation object is implicitly addressed by the service.

6.3.12.3 Fragmentation service specification

6.3.12.3.1 M_UNITDATA

For the definition of this service see IEEE 802.3.

This local service shall be used to transmit (M_UNITDATA.req) or receive (M_UNITDATA.ind) a PDU.

6.3.12.4 Invocation of the Fragmentation object

For the invocation of the Fragmentation object the following rules apply:

- Fragmentation objects shall exist for each interface.

6.3.13 IP suite ASE

6.3.13.1 Overview

The utilization of the IETF RFC 768 (UDP), IETF RFC 791 (IP), IETF RFC 792 (ICMP), IETF RFC 826 (ARP), IETF RFC 1112 (IP Multicasting) standards are defined for this specification. It includes definitions for UDP ports and IP multicast addresses and utilization of IP header fields for RT over UDP.

This ASE shall contain current volatile values of operation. Permanent values in non volatile memory shall be a matter of the DCP ASE defined in 6.3.1.

6.3.13.2 IP suite class specification

6.3.13.2.1 General

The IP suite ASE defines one Physical Device Management object type:

6.3.13.2.2 Template

A IP suite object is described by the following template:

ASE: IP suite ASE

CLASS: IP suite

CLASS ID: not used

PARENT CLASS: top

ATTRIBUTES:

1	(m)	Key Attribute: Implicit
2	(m)	Attribute: IP
2.1	(m)	Attribute: IP Address
2.2	(m)	Attribute: Subnet Mask
2.3	(m)	Attribute: Standard Gateway
3	(m)	Attribute: ARP
3.1	(m)	Attribute: Cache Size
3.2	(m)	Attribute: Cache Timeout
3.3	(m)	Attribute: List of Cache Entries
3.3.1	(m)	Attribute: IEEE 802.3 MAC Address
3.3.2	(m)	Attribute: IP Address
4	(m)	Attribute: List of IP Multicast addresses
4.1	(m)	Attribute: IP Multicast address

SERVICES:

1	(o)	OpsService: Set ARP Cache

6.3.13.2.3 Attributes

Implicit

The key attribute Implicit indicates that the IP suite object is implicitly addressed by the service.

IP

This attribute contains the following attributes:

IP Address

This attribute contains the current IP address according to IETF RFC 791.

Attribute type: Unsigned32

Subnet Mask

This attribute contains the current subnet mask according to IETF RFC 791.

Attribute type: Unsigned32

Standard Gateway

This attribute contains the current IP address of the standard gateway according to IETF RFC 791.

Attribute type: Unsigned32

ARP

This attribute contains the following attributes:

Cache Size

This attribute contains the size of the ARP cache and shall contain a value to cache all devices that the device is connected to. IEC 61784-2 defines a recommended value.

NOTE 1 The cache size influences the performance of the IO operation.

Attribute type: Unsigned32

Cache Timeout

This attribute contains the timeout value for the refresh of the ARP cache entry (see IETF RFC 826). This attribute influences the time to recognize changed IP address / MAC address couples in case of device replacement. IEC 61784-2 defines a recommended value.

NOTE 2 Static ARP cache entries for IO devices provide a suitable solution if supported by the UDP/IP stack.

Attribute type: Unsigned32

List of Cache Entries

This attribute list contains the following attributes:

IEEE 802.3 MAC Address

This attribute contains the value of the IEEE 802 MAC address corresponding to the IP address.

Attribute type: OctetString[6]

IP Address

This attribute contains the IP address according to IETF RFC 791.

Attribute type: Unsigned32

List of IP Multicast addresses

This attribute list contains the following attributes:

IP Multicast address

This attribute contains the IP Multicast address used for multicast communication relation. It shall be set according IETF RFC 2365 and shall follow the building rule in relation to the Frame ID shown in Table 97.

Attribute type: Unsigned32

Allowed values with associated MAC address and Frame ID are listed in Table 97.

Table 97 – IP Multicast address

IP multicast address	Associated Multicast MAC address	Associated Frame ID	Usage
239.192.248.0	01-00-5E-40-F8-00	0xF800	Used for multicast communication relations in conjunction with RT_CLASS_UDP
239.192.248.1 to 239.192.251.254	01-00-5E-40-F8-01 to 01-00-5E-40-FB-FE	0xF801 to 0xFBFE	Used for multicast communication relations in conjunction with RT_CLASS_UDP
239.192.251.255	01-00-5E-40-FB-FF	0xFBFF	Used for multicast communication relations in conjunction with RT_CLASS_UDP

6.3.13.3 IP suite service specification**6.3.13.3.1 Set ARP Cache**

This local service may be used to manipulate the local ARP cache loading entries. Table 98 shows the parameters of the service.

Table 98 – Set ARP Cache

Parameter name	req	cnf
Argument	M	
Number of Entries	M	
List of Cache Entries	U	
IEEE 802.3 MAC Address	M	
IP Address	M	
Result(+)		S(=)
Result(-)		S(=)

Argument

The argument shall convey the service specific parameters of the service request.

Number of Entries

This optional parameter list shall contain the number of Cache Entries that follows. The number shall not exceed the value of the attribute Cache Size.

List of Cache Entries

The optional parameter list shall consist of the following parameter:

IEEE 802.3 MAC Address

This parameter contains the value for the equivalent ASE attribute.

IP Address

This parameter contains the value for the equivalent ASE attribute.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

6.3.14 Domain name system ASE

6.3.14.1 Overview

The utilization of the IETF RFC 1034 (DNS) standard is defined for this specification. It includes the usage and the content syntax of domain name.

6.3.14.2 DNS class specification

6.3.14.2.1 General

The DNS ASE defines one DNS object type:

6.3.14.2.2 Template

A DNS object is described by the following template.

ASE: DNS ASE

CLASS: DNS

CLASS ID: not used

PARENT CLASS: top

ATTRIBUTES:

- 1 (m) Key Attribute: Implicit
- 2 (m) Attribute: DNS
- 2.1 (m) Attribute: Preferred DNS Server
- 2.2 (m) Attribute: Alternate DNS Server 1
- 2.3 (m) Attribute: Alternate DNS Server 2
- 2.4 (m) Attribute: Alternate DNS Server 3

SERVICES:

- 1 (a) OpsService: DNS Get Host By Name

6.3.14.2.3 Attributes

Implicit

The key attribute Implicit indicates that the DNS object is implicitly addressed by the service.

DNS

This attribute contains the following attributes:

Preferred DNS Server

This attribute contains the IP address of the preferred DNS server according to IETF RFC 791.

Attribute type: Unsigned32

Alternate DNS Server 1

This attribute contains the IP address of the first alternate DNS server according to IETF RFC 791.

Attribute type: Unsigned32

Alternate DNS Server 2

This attribute contains the IP address of the second alternate DNS server according to IETF RFC 791.

Attribute type: Unsigned32

Alternate DNS Server 3

This attribute contains the IP address of the third alternate DNS server according to IETF RFC 791.

Attribute type: Unsigned32

6.3.14.3 DNS service specification

The DNS services shall be derived from the IETF RFC 1034 (DNS) standard.

6.3.15 Dynamic host configuration ASE**6.3.15.1 Overview**

The utilization of the IETF RFC 2131 (DHCP) standard is defined for this specification. It includes the usage and the content syntax of Client ID.

6.3.15.2 DHCP class specification**6.3.15.2.1 General**

The DHCP ASE defines one DHCP object type:

6.3.15.2.2 Template

A DHCP object is described by the following template:

ASE:	DHCP ASE
CLASS:	DHCP
CLASS ID:	not used
PARENT CLASS:	top
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
SERVICES:	
1	(o) OpsService: DHCP Get IP

6.3.15.2.3 Attributes**Implicit**

The key attribute Implicit indicates that the DHCP object is implicitly addressed by the service.

6.3.15.3 DHCP service specification

The DHCP services shall be derived from the IETF RFC 2131 (DHCP) standard.

6.3.16 Simple network management ASE**6.3.16.1 Overview**

The utilization of the IETF RFC 2674 (Bridges with traffic classes), IETF RFC 2737 (MIB 2), IETF RFC 2863 (IF MIB), IETF RFC 3418 (SNMP), IETF RFC 3621 (Power over Ethernet MIB), IETF RFC 4836 (MAU MIB) standards are defined for this specification. It includes the usage of different MIBs. Furthermore, an enhancement of the LLDP MIB is specified in IEC 61158-6-10.

6.3.16.2 SNMP class specification

6.3.16.2.1 General

The SNMP ASE defines one SNMP object type:

6.3.16.2.2 Template

A SNMP object is described by the following template:

ASE: **SNMP ASE**

CLASS: **SNMP**

CLASS ID: **not used**

PARENT CLASS: **top**

ATTRIBUTES:

- 1 (m) Key Attribute: Implicit
- 2 (m) Attribute: Type10 MIB
- 2.1 (m) Attribute: Enterprise number
- 2.2 (o) Attribute: List of Vendors
- 2.2.2 (m) Attribute: Vendor OUI
- 3 (m) Attribute: List of Interfaces
- 3.1 (m) Attribute: List of Ports
- 3.1.1 (m) Attribute: ifInOctets
- 3.1.2 (m) Attribute: ifOutOctets
- 3.1.3 (m) Attribute: ifInDiscards
- 3.1.4 (m) Attribute: ifOutDiscards
- 3.1.5 (m) Attribute: ifInErrors
- 3.1.6 (m) Attribute: ifOutErrors

SERVICES:

- 1 (o) OpsService: SNMP Get

6.3.16.2.3 Attributes

Implicit

The key attribute Implicit indicates that the SNMP object is implicitly addressed by the service.

Type10 MIB

This attribute is composed of the following elements:

Enterprise number

This attribute contains the enterprise number as MIB identifier. The value shall be according to Table 99.

Table 99 – Enterprise number

Value	Meaning
24 686	Enterprise number for Type10 MIB

List of Vendors

This list attribute consists of the following attributes:

Vendor OUI

This attribute defines which additional vendor specific MIB extension exists.

Attribute type: Unsigned32

The allowed values are shown in Table 100.

Table 100 – Vendor OUI

Value	Meaning
Type10 OUI	Type10 defines subsequent structured data
Vendor OUI	Vendor defines subsequent structured data
Others	Reserved

List of Interfaces

This list attribute consists of the following attributes:

List of Ports

This list attribute consists of the following attributes:

NOTE Note that the value “0” for the particular statistic counters is blurry. If the attribute has the value “0”, the statistic counter may not be supported or else there was nothing counted up to now. If however the counters ifInOctets and ifOutOctets are returned with the value “0”, one may derive that also the other counters, if being “0”, are unsupported.

ifInOctets

This attribute contains the number of octets in valid MAC frames received on this interface, including the MAC header and FCS. For details see IETF RFC 2020.

Attribute Type: Unsigned32

Allowed values: 0 (meaning the statistic counter is not supported), 0 to 0xFFFFFFFF (meaning the number of octets)

ifOutOctets

This attribute contains the number of octets transmitted in valid MAC frames on this interface, including the MAC header and FCS. For details see IETF RFC 1573.

Attribute Type: Unsigned32

Allowed values: 0 (meaning the statistic counter is not supported), 0 to 0xFFFFFFFF (meaning the number of octets)

ifInDiscards

This attribute contains the number of Rx packets which were chosen to be discarded due to resource conflicts in upper instances. For details see IETF RFC 1573.

Attribute Type: Unsigned32

Allowed values: 0 (meaning the statistic counter is not supported), 0 to 0xFFFFFFFF (meaning the number of Rx packets)

ifOutDiscards

This attribute contains the number of Tx packets which were chosen to be discarded due to resource conflicts in instances like an integrated switch. For details see IETF RFC 1573.

Attribute Type: Unsigned32

Allowed values: 0 (meaning the statistic counter is not supported), 0 to 0xFFFFFFFF (meaning the number of Tx packets)

ifInErrors

This attribute contains the sum for this interface of bad received frames. For details see IETF RFC 2020.

Attribute Type: Unsigned32

Allowed values: 0 (meaning the statistic counter is not supported), 0 to 0xFFFFFFFF (meaning the sum of bad received frames)

ifOutErrors

This attribute contains the transmit errors on this interface. For details see IETF RFC 1573.

Attribute Type: Unsigned32

Allowed values: 0 (meaning the statistic counter is not supported), 0 to 0xFFFFFFFF (meaning the transmit errors)

6.3.16.3 SNMP service specification

The SNMP services shall be derived from the IETF RFC 3418 (SNMP) standard.

6.3.17 Common DL mapping ASE

6.3.17.1 Overview

This ASE provides a common interface for DL mapping.

6.3.17.2 DL Mapping class specification

6.3.17.2.1 General

The DL Mapping ASE defines one DL Mapping object type:

6.3.17.2.2 Template

A DL Mapping object is described by the following template:

ASE: DL Mapping ASE
CLASS: DL Mapping
CLASS ID: not used
PARENT CLASS: top
ATTRIBUTES:
 1 (m) Key Attribute: Implicit
SERVICES:
 1 (m) OpsService: P Data
 2 (m) OpsService: N Data
 3 (m) OpsService: A Data
 4 (m) OpsService: C Data

6.3.17.2.3 Attributes

Implicit

The key attribute Implicit indicates that the DL Mapping object is implicitly addressed by the service.

6.3.17.3 DL Mapping service specification

6.3.17.3.1 P Data

This service is used to transfer time stamped data from the IO device to the IO controller and vice versa. Table 101 shows the parameters of the service.

Table 101 – P Data

Parameter name	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	

Parameter name	req	ind	cnf
D Port	M	M(=)	
T Stamp		M	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
N SDU	M	M(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter is the local identifier for the desired CR.

D Port

This parameter contains the desired Port ID.

T Stamp

This parameter contains the time stamp.

DA

This parameter contains the destination address according to IEEE 802.3.

SA

This parameter contains the source address according to IEEE 802.3.

VLAN Prio

This parameter contains the priority according to IEEE 802.1Q.

VLAN ID

This parameter contains the VLAN ID according to IEEE 802.1Q.

LT

This parameter contains the priority according to IEEE 802.3.

N SDU

This parameter contains the APDU.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Status

This parameter contains the reason of the error.

6.3.17.3.2 N Data

This service is used to transfer non RTA and RTC data from the IO device to the IO controller and vice versa. Table 102 shows the parameters of the service.

Table 102 – N Data

Parameter name	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
N SDU	M	M(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter is the local identifier for the desired CR.

D Port

This parameter contains the desired Port ID.

DA

This parameter contains the destination address according to IEEE 802.3.

SA

This parameter contains the source address according to IEEE 802.3.

VLAN Prio

This parameter contains the priority according to IEEE 802.1Q.

VLAN ID

This parameter contains the VLAN ID according to IEEE 802.1Q.

LT

This parameter contains the priority according to IEEE 802.3.

N SDU

This parameter contains the APDU.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Status

This parameter contains the reason of the error.

6.3.17.3.3 A Data

This service is used to transfer RTA data from the IO device to the IO controller and vice versa. Table 103 shows the parameters of the service.

Table 103 – A Data

Parameter name	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
A SDU	M	M(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter is the local identifier for the desired CR.

D Port

This parameter contains the desired Port ID.

DA

This parameter contains the destination address according to IEEE 802.3.

SA

This parameter contains the source address according to IEEE 802.3.

VLAN Prio

This parameter contains the priority according to IEEE 802.1Q.

VLAN ID

This parameter contains the VLAN ID according to IEEE 802.1Q.

LT

This parameter contains the priority according to IEEE 802.3.

A SDU

This parameter contains the APDU.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Status

This parameter contains the reason of the error.

6.3.17.3.4 C Data

This service is used to transfer RTC data from the IO device to the IO controller and vice versa. Table 104 shows the parameters of the service.

Table 104 – C Data

Parameter name	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
N SDU	M	M(=)	
APDU Status	M	M(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument

The argument shall convey the service specific parameters of the service request.

CREP

This parameter is the local identifier for the desired CR.

D Port

This parameter contains the desired Port ID.

DA

This parameter contains the destination address according to IEEE 802.3.

SA

This parameter contains the source address according to IEEE 802.3.

VLAN Prio

This parameter contains the priority according to IEEE 802.1Q.

VLAN ID

This parameter contains the VLAN field according to IEEE 802.1Q.

LT

This parameter contains the priority according to IEEE 802.3.

C SDU

This parameter contains the APDU.

APDU Status

This parameter contains the field APDU Status.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Status

This parameter contains the reason of the error.

7 Communication model for decentralized periphery

7.1 Concepts

7.1.1 User requirements

The typical automation system consists of one or several programmable controllers connected with IO-systems to the machine/process. Today, an IO-system is mostly implemented as a hierarchical decentralized system or a set of single point connections with a well defined electrical interface such as 24 V digital signals or 4 mA to 20 mA analogue signals.

This communication model defines a decentralized system and the connection is based on IEEE 802.3 information and telecommunication technology. The defined object model is an enhancement of the IEC 61158-5-3 object model.

7.1.2 Features

Combining the structure defined in IEC 61131-1 the following advantages are achieved:

- Changes in the plant cause only small changes in wiring.
- Critical signals can be transferred over short distances to the remote field device (sensor/actuator) and to/from the programmable controller over long distances with high accuracy.
- Parameterization and diagnosis can be done without additional wiring.
- Commissioning of subparts without additional installations.

Table 105 gives an overview of the requirements and features of the system.

Table 105 – Requirements and features

Requirement	Feature
Short reaction time	Cyclic exchange of more than 1 000 inputs and outputs with 32 field devices in less than 1 ms
Automation system consists of one or several programmable controller	Mono-controller or Multi-controller operation, more than one IO controller can be associated to one IO device
Simple field device	Simple protocol, low cost communication interface
Adaptation of application model of IEC 61158-5-3 / IEC 61158-6-3	Similar application process object model with compatibility modus
Unified protocols	Use of the same real-time transport protocol mechanism for different application models

Requirement	Feature
Excellent diagnosis	Diagnosis and alarm object model for IO devices
Smart field devices	Acyclic communication which provides a flexible and enhanced addressing scheme of data within IO devices. The possibility to transfer an alarm from the IO device to the IO controller and vice versa with an explicit acknowledgement.
Efficient communication between IO devices and between IO controller	Provider/Consumer mechanism
Synchronization of applications	Isochronous mode, jitter less than 1 μ s, signaling to the application
Synchronization of clocks	High precision time synchronization
Redundancy	IO device and IO controller redundancy, support for media redundancy
Interoperability	Precise and complete definitions including the definition of the system behavior
Modification during operation	Dynamic reconfiguration

7.1.3 Associations

The communication model supports three types of associations which are characterized by device types involved in the association.

NOTE 1 Additional associations e.g. for upload or download may be possible but are not part of the communication model.

The communication model supports the communication of field devices with one or more controlling devices (e.g. programmable controller or distributed control system) via associations (see Figure 9). It includes cyclic data exchange of IO data and transmission of alarms, transmission of parameter data, configuration data, identification data, LogBook data, diagnosis data, and record data.

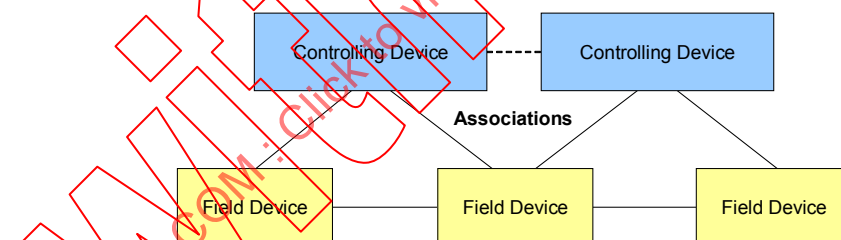


Figure 9 – Example of communication between controlling devices and field devices

Furthermore, the communication between an engineering device and several controlling or field devices is also supported (see Figure 10). The association between engineering and field devices includes transmission of parameter data, configuration data, identification data, LogBook data, diagnosis data, and record data via connectionless RPC protocol.

NOTE 2 The association between engineering and controlling device includes transmission of controller diagnosis, domains and commands for device activation via connectionless RPC protocol (for further studies).

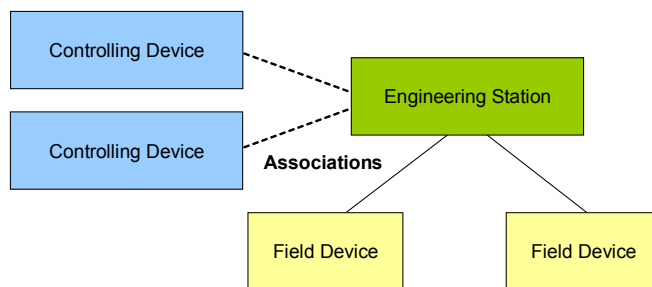


Figure 10 – Example of communication between an engineering station and several controlling and field devices

In addition to these communication models, it also supports the communication between a server station and field devices (see Figure 11). The association between a server station and field device includes transmission of record data via connectionless RPC protocol.

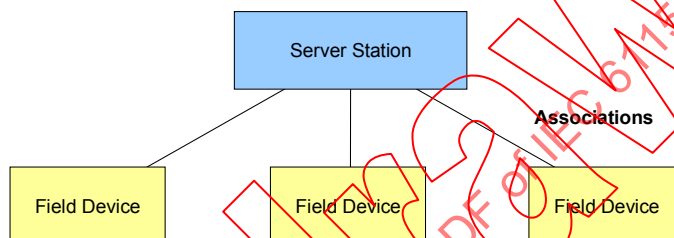


Figure 11 – Example of communication between field devices and a server station

In addition to these communication models, it also supports the communication between field devices (see Figure 12). This communication model includes a controlling device to setup and control the association between the field devices.

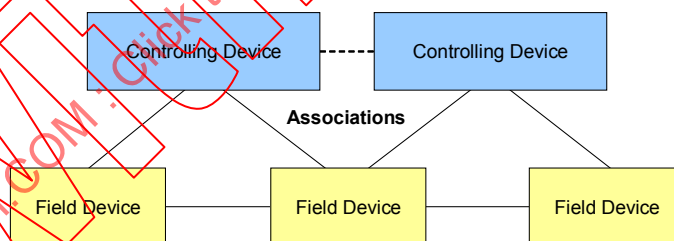


Figure 12 – Example of communication between field devices

NOTE 3 The association between controlling devices may include e.g. the transmission of IO data, diagnosis, alarms and parameter.

7.1.4 Device types

7.1.4.1 General

A physical automation device may host one or more of the same or different device types with appropriate associations.

7.1.4.2 IO controller

The IO controller is a controlling device which is associated with one or more IO devices (field devices). The IO controller performs one or more of the following functionalities:

- a) cyclic functionalities

- exchange of IO data with related IO devices
- b) acyclic functionalities
 - read diagnosis from IO devices
 - configuration of IO devices
 - writes parameter data to IO devices (startup or application parameter)
 - treatment of configuration and diagnosis requests of an engineering device
 - establish context to IO devices by means of context management
 - acyclic access to record data of IO devices
 - treatment of alarms from IO devices
 - sending of alarms to IO devices
- c) general functionalities
 - redundancy
 - dynamic reconfiguration
 - isochronous operation
 - provider/consumer communication between IO devices and IO controllers
 - clock synchronization

7.1.4.3 IO supervisor

The IO supervisor is an engineering device which manages provision of configuration data (parameter sets) and collection of diagnosis data from a IO controllers and/or IO devices.

7.1.4.4 IO parameter server

The IO parameter server is a server station which serves to store and to load application configuration data (record data objects) from IO devices (client).

7.1.4.5 IO device

7.1.4.5.1 General

The IO device is a field device and performs the following activities depending on the functionality.

The functionalities are:

- a) cyclic functionalities
 - exchange of IO data with one or more assigned IO controller
 - exchange of IO data with related IO devices
- b) acyclic functionalities
 - provisions for diagnosis data supply
 - treatment of configuration requests of an IO controller
 - provision of acyclic access to record data for an IO controller
 - provision of alarms to the assigned IO controller
 - treatment of parameters (startup or application parameter)
 - treatment of configuration and diagnosis requests of an engineering device
 - treatment of alarms from IO controllers
 - sending of alarms to IO controllers
- c) general functionalities

- provider/consumer communication between IO devices and IO controllers
- redundancy
- dynamic reconfiguration
- isochronous operation
- clock synchronization

IO device conformance classes are defined to support common functions.

In general, an IO device is hierarchical composed including the following elements:

- one or more IO device instances (see Annex A);
- each IO device instance includes one or more application processes referenced by its identifier (API);
- each API includes one or more slots;
- each slot includes one or more subslots;
- each subslots may include one or more channels.

The IO device is composed of different structural units to group application process objects and provides a certain level of abstraction. These structural units may reflect hardware components or virtual functional units of the field device. The aim is to provide a suitable set of address parameter. These structural units are referred to as slots, subslots, and channels which may reflect also physical units, sub-units, or a single connection point of an IO device. Figure 13 shows the structure for one arbitrary API. The structural units instance, API, slot and subslot are accessed by means of Object UUID, API Number, Slot Number and Subslot Number within the address model.

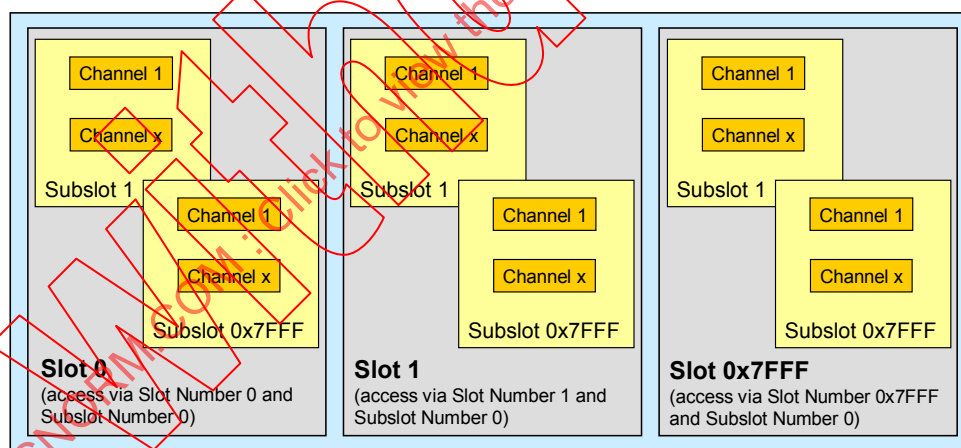


Figure 13 – Structural units of one arbitrary API of an IO device (general)

Furthermore, subslots 0 to 0x7FFF may be used in all APIs of an IO device in parallel. A particular application relationship may be established to more than one application processes. An IO device instance with more than one Application Process Identifier referred to as Multi-API-Instance.

NOTE An IO device instance supporting a particular profile API is a Multi-API-Instance because the API 0 support of the IO device instance is always mandatory.

The Subslot Number 0 shall be used to represent the module in the special case “pull module”, if the “pull module” alarm is not supported. In this case, the “pull module” alarm is simulated with a “pull submodule” alarm on subslot 0. Then it is not possible to have a “real” Subslot 0 containing IO channels, diagnosis or record data.

If the “pull module” alarm is supported, a “real” Subslot 0 exists, and Subslot 0 may contain IO channels, diagnosis or record data.

Additionally, special subslots in the range from 0x8000 to 0x8FFF are defined for API 0. An IO device shall contain at least one interface with ports. Therefore, at least one Slot (e.g. Slot Number 0), shall contain up to 16 further special Subslots from 0x8000 (Interface 1), 0x8100 (Interface 2), to 0x8F00 (Interface 16) referred to as Interface Subslots. Interface Subslots define the remote access to IP address parameters and the name of the interface (DNS name). Furthermore, each Interface Subslot shall have at least one and up to 255 assigned ports. These subslots from 0x8i01(Port 1) to 0x8iFF (Port 255) referred to as Port Subslots. The subslot defines the remote access to port specific parameter e.g. Own Port ID. Figure 14 depicts an example with several ports on the same slot belonging to one interface. All these special subslots shall only be used within API 0. The values “port-001” to “port-255” shall be used for naming the port submodules if all port submodules belonging to this interface are located in the same slot as the interface submodule.

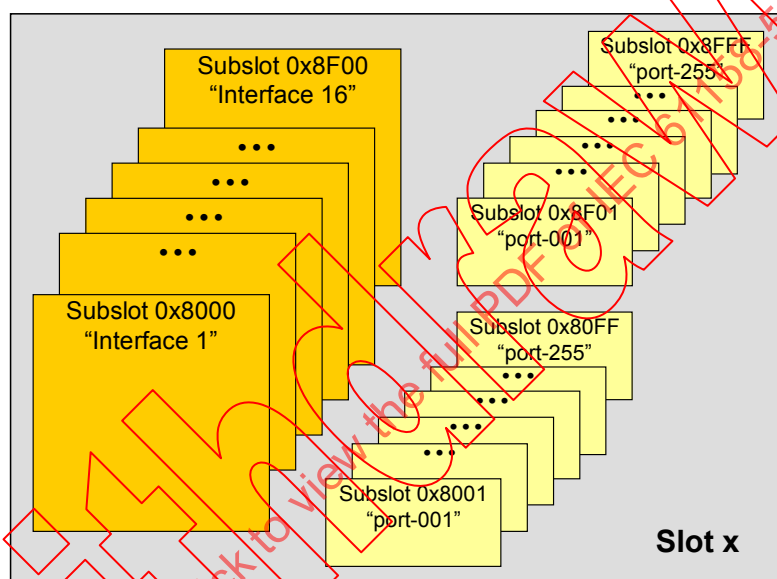


Figure 14 – Example 1 structural units for interfaces and ports within API 0

Figure 15 depicts an example where several port submodules belonging to an interface are located on several slots. The values “port-001-xxxxx” to “port-255-zzzzz” shall then be used for naming the port submodules, whereas “xxxxx” denoted the slot number with leading “0” (e.g. port-002-00005”).

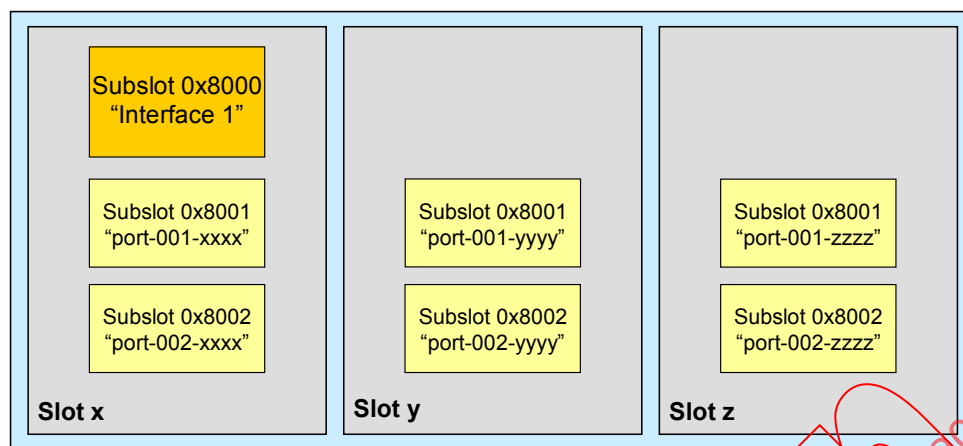


Figure 15 – Example 2 structural units for interfaces and ports within API 0

All possible configurations of an IO device should be described in the GSD file of the device.

In the special case system redundancy the subslot numbering scheme for interfaces and ports is extended. In this special case only

- Subslot number 0x8000 to 0x8FFF indicate a subslot used for interface or port on the left or below delivering node.
- Subslot number 0x9000 to 0x9FFF indicate a subslot used for interface or port on the right or above delivering node.

For the subslot number range 0x9000 to 0x9FFF the same rules apply as for range 0x8000 to 0x8FFF; i.e. 0x9i00 for an interface submodule with “i” in the range from 0 to 15, denominating the number of the interface, 0x9ipp for a port submodule with “pp” in the range from 1 to 255, denominating the number of the port.

7.1.4.5.2 Slot

The application layer uses slots to reflect the structuring of functions or components (e.g. hardware modules, logical units) inside an IO device. A slot shall have one or more subslots which represent the further level to structure the data.

Each slot shall be addressed by a slot number starting with zero. The last possible Slot Number is 0x7FFF. Numbering may contain gaps.

NOTE The slot model is an abstract address model via numeric references. Real hardware modules or virtual functional units may even occupy more than one slot. The device characteristics concerning use of slots are defined in a general station description for configuration purpose.

Furthermore, each module owns configuration data. Within the connection establishment phase the requested configuration shall be compared with the real configuration data. If a slot is not occupied with a physical or virtual module then it is not part of the configuration.

In modular IO devices slots determine the actual configuration of the field device. The number of slots is fixed according to the device specification.

A compact IO device is a special case of a modular IO device with one or more fixed configurations.

7.1.4.5.3 Subslot

The application layer uses subslots to reflect the structuring of functions or further components (e.g. hardware units, logical units) inside a slot of an IO device. A subslot may

have one or more channels which represent the real structure of the Input and/or Output Data. These channels may be a further subdivision of the Input and/or Output Data object.

Each subslot will be addressed by a subslot number 0 to 0x7FFF. Furthermore, within a slot special subslots 0x8000 to 0x8FFF for interfaces and related ports may exist; in the special case system redundancy subslots 0x9000 to 0x9FFF for interface and related ports on the right or above delivering endnode may exist.

If ARProperties.PullModuleAlarmAllowed is not used, subslot number zero is used in conjunction with AlarmType “Pull” to address a certain module within the IO device. In this case, it shall not contain channels and IO data.

If ARProperties.PullModuleAlarmAllowed is used, subslot number zero acts as a real subslot. On the other hand, real subslots may contain channels and IO data. The actual use of subslot number and index to address data within the device is manufacturer specific.

Subslot 0 is addressing the module within the scope of the Pull Alarm to signal pulling of a module, if ARProperties.PullModuleAlarmAllowed is not used. If ARProperties.PullModuleAlarmAllowed is used, Subslot 1 to 0x7FFF is freely usable within the scope of the addressed AP.

Within a slot, a subslot provides one or several of the following objects:

- Record Data (data with open semantics, e.g. parameter), addressed by
 - slot number (0 to 0x7FFF),
 - subslot number (0 to 0x9FFF or else 1 to 0x9FFF, according to ARProperties.PullModuleAllowed), and
 - index (0 to 0xFFFF).
- IO Data as an IO combined data structure, whereas the structure is depending on general station description, each IO data structure includes status information, and addressed by
 - slot number (0 to 0x7FFF), and
 - subslot number (0 to 0x7FFF or else 1 to 0x7FFF, according to ARProperties.PullModuleAllowed)

NOTE The special subslots used for interface and related ports do not contain IO data.

- Diagnosis contains the channel or generic diagnosis for further identification, and addressed by
 - slot number (0 to 0x7FFF),
 - subslot number (0 to 0x9FFF or else 1 to 0x9FFF, according to ARProperties.PullModuleAllowed), and
 - index (implicitly set by the diagnosis service),
- Alarms, addressed by
 - slot number (0 to 0x7FFF),
 - subslot number (0 to 0x9FFF or else 1 to 0x9FFF, according to ARProperties.PullModuleAllowed), and
- alarm type.

7.1.4.5.4 Channel

The application layer uses slots and subslots to reflect the structuring of functions and components (e.g. hardware units, logical units) inside an IO device. A slot/subslot may have one or more channels which represent the real structure of the Input and/or Output Data. These channels may be a further subdivision of the Input and/or Output Data object.

Channels are identified in diagnosis or alarm information only.

7.1.4.5.5 Identification hierarchy

The following identification hierarchy is defined for the application layer elements:

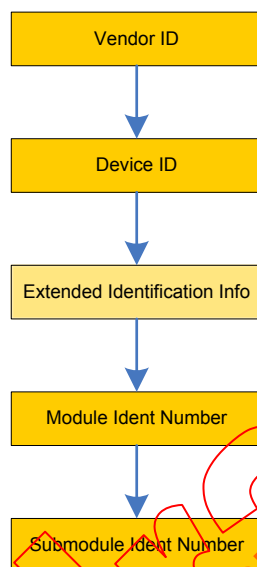


Figure 16 – Identification hierarchy

A particular IO device is identified over Vendor ID and Device ID. The Vendor ID is a globally organizational administered number identifying a particular vendor; the Device ID is uniquely assigned to the device type by the vendor of the device. See also 7.6.1.

The Extended Identification Info allows extending the identification rules for the common profile “Controller to Controller Communication”. The Extended Identification Info is provided as an AR Vendor Block during the Context ASE’s service Connect. If no extended identification info is provided, no additional identification rules apply.

The Module Ident Number shall uniquely designate all modules which may appear in a specific device. If a module can be placed in different devices, the ModuleIdentNumber shall be assigned uniquely across all these devices.

The Submodule Ident Number shall uniquely designate all submodules which may appear in a specific module. Beyond identifying the submodule’s hardware, it also identifies the representation of the submodule’s input and/or output data regarding its length and data type. If a submodule can be placed in different modules, the SubmoduleIdentNumber shall be assigned uniquely across all these modules.

If the same hardware submodule has different representations of input and/or output data, the submodule shall have different Submodule Ident Numbers for each of those representations. Note that each of these Submodule Ident Numbers shall still be able to identify the specific submodule, so each of these ident numbers shall be different from the ident numbers of other submodule hardware on the same module.

So, while the Vendor ID, Device ID and Module Ident Number shall be one-to-one, i.e. there is exactly one ident number for a particular hardware type, a submodule may have several unique SubmoduleIdentNumbers.

The identification numbers are stored in the hardware and can be read out, e.g. via the RPC Object-UUID (Vendor ID and Device ID), via I&M functions (Vendor ID and Device ID), via AR Data (Extended Identification Info) and via Real Identification Data (Module Ident Number and Submodule Ident Number).

7.1.4.5.6 Combined Object

An IO device may define so called “combined objects”. A particular combined object is a group of related submodules, e.g. an interface and its ports. One of the submodules is defined as the combined object’s “access point”, e.g. the interface.

Every submodule of a particular combined object shall be part of the application relation.

A combined object will be parameterized by writing the CO Container to its access point; addressed by API, slot, subslot and index. The CO Container holds the parameterization for all submodules; for this it is substructured in subblocks; each having an address for the submodule (API, slot and subslot), an index for the data record to write and the data itself.

The access point distributes the Record Data objects stored in the CO Container to the submodules of the combined object, when being parameterized. If some submodule is not available at this time, the access point keeps the data and distributes it to the submodule, if this gets available. If a particular submodule is plugged for example, it informs the IO controller as usual by a Plug alarm; however since the IO controller only has parameterization information for the combined object access point (i.e. the CO Container), the parameterization for this submodule from the IO controller is empty.

This procedure allows plugging a bunch of submodules at the same time (e.g. a pluggable interface with its ports) and having an implicit parameterization bracket for all submodules together (this is especially important in the context of the physical device). This however works only if all record data for the combined object is placed into the combined object container.

A combined object has no impact to the IO controller, from his point of view only the CO access point owns parameters; there are some submodules with and some without parameters.

Record data is always read directly from the particular submodule; in particular the CO Container is not readable.

A special case for the combined object is used for the physical device (e.g. in the context of pluggable interfaces and ports). The CO access point in this case is the interface submodule. The CO Container contains the parameter for all interface and port submodules.

7.1.4.5.7 Identification and maintenance data

This data identifies a submodule and provides additional information to support the end user during various scenarios of a submodules life cycle be it configuration, commissioning, parameterization, diagnosis, maintenance, repair, firmware update, asset management, audit trailing, and alike.

For this an orderable unit is the focus. An orderable unit may however consist of multiple submodules (e.g. a module consisting of an interface submodule, a bunch of port submodules, and further submodules). Such an orderable unit has just one distinct identification and maintenance data.

Exactly one of the orderable unit’s submodules is responsible for carrying the identification and maintenance data. This submodule is called the carrier. Only the carrier allows write access to the identification and maintenance data.

To allow each application relation to access the identification and maintenance information of a particular submodule (e.g. shared device), each submodule shall support reading identification and maintenance data. It thus acts as a proxy to the carrier and returns the identification and maintenance information as stored at the carrier.

The following rules apply to I&M filter data (see 7.3.6.3.18):

- The list of orderable units is returned with I&M0 carrier data; in this list the submodules being carrier of I&M data are listed.
- There is at least one submodule, where I&M1, I&M2, I&M3 and I&M4 can be written. This submodule is indicated in the device representative list.

7.1.5 Instance model and device addresses

The design and construction of physical automation devices is beyond the scope of this specification and therefore vendor specific. However, Annex A provides a model for physical devices to host one or more IO controller or IO devices.

7.1.6 Application process

7.1.6.1 Overview

In the application process environment, an application may be partitioned and distributed to a number of devices on the network. Each of these partitions is referred to as an application process (AP). A device may have several APs as shown in Figure 17. In this case each individual AP is uniquely identified by an AP Identifier (API).

Each IO device shall support a Default AP with API = 0.

The Default APs shall provide device related information. Other APs are reserved to be uniquely assigned to device profiles and further standardized use.

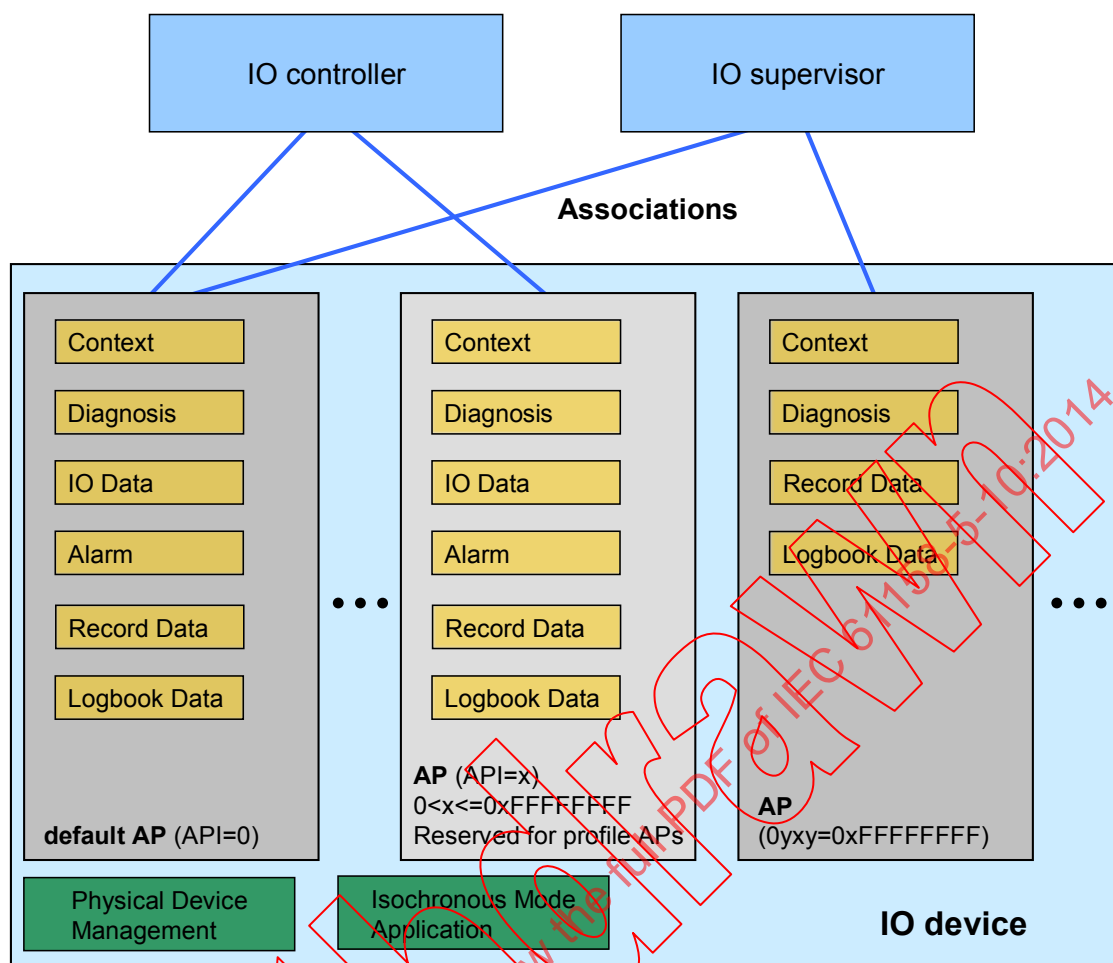


Figure 17 – Overview of application processes

An AP may be distributed to several slots and subslots. Figure 18 shows the relationship between the APs, the data elements, the slots, and subslots of an IO device. The gray boxes illustrate that there exist no “real” subslot 0. Subslot 0 shall also not contain e.g. IO data, as shown in Figure 18.

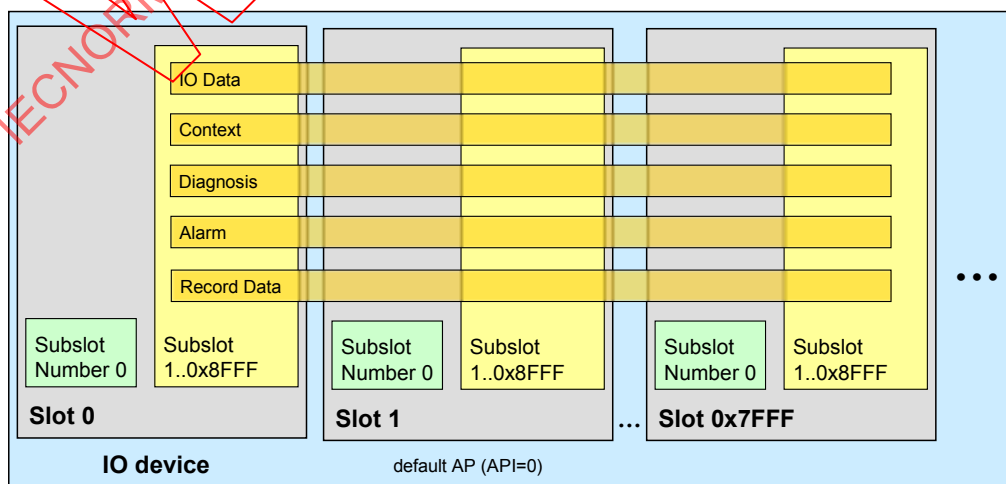


Figure 18 – IO device with APs, slots and subslots

The Subslot 0x8000 to 0x9FFF shall exist only in API 0.

NOTE 1 The mapping of a real object to the addressing model is a local matter. It means that one real object can be mapped or grouped in different APs and subslots. However, from the network perspective all address information API/slot/subslot are unique. Each API/slot/subslot can provide a different but consistent view on network visible objects. However, the IO device avoids conflicts concerning multiple use of output data referred to as overlapping outputs. An application association from a supervisor is always able to take control over output data and the controller device is informed about this by means of a special alarm.

The following rules shall be applied for the AP:

- Device instance identification data as part of the Context ASE shall always be identical regardless of the addressed AP
- API 0 addresses the application process
- API 1 to 0xFFFFFFFF is reserved for IO profiles which shall be unique.

Besides the vendor and profile specific use of the address model the following rules shall be applied for the Index:

- The indices 0 to 0x7FFF shall be used by the device manufacturer for user specific Record Data objects within the scope of the addressed AP. For API 0 indexes are vendor specific. For other APs further definitions specified in the related profile shall be applied.

NOTE 2 Index 0-0xFF can be used for migration of IEC 61784-1 CP3/1 conform devices to access the same Process Data Objects with same index.

- The indices 0x8000 to 0xAFFF are reserved for IEC 61158-6-10 use. The scope of the Record Data object is a particular subslot within the scope of the addressed AP.
- The indices 0xB000 to 0xBFFF are reserved for profile use. The scope of the Record Data object is a particular subslot within the scope of the addressed AP.
- The indices 0xC000 to 0xCFFF are reserved for IEC 61158-6-10 use. The scope of the Record Data object is a particular slot within the scope of the addressed AP.
- The indices 0xD000 to 0xDFFF are reserved for profile use. The scope of the Record Data object is a particular slot within the scope of the addressed AP.
- The indices 0xE000 to 0xEBFF are reserved for IEC 61158-6-10 use. The scope of the Record Data object is a particular application relation.
- The indices 0xEC00 to 0xEFFF are reserved for profile use. The scope of the Record Data object is a particular application relation.
- The indices 0xF000 to 0xF3FF are reserved for IEC 61158-6-10 use. The scope of the Record Data object is a particular AP.
- The indices 0xF400 to 0xF7FF are reserved for profile use. The scope of the Record Data object is a particular AP.
- The indices 0xF800 to 0xFBFF are reserved for IEC 61158-6-10 use. The scope of the Record Data object is the device.
- The indices 0xFC00 to 0xFFFF are reserved for profile use. The scope of the Record Data object is the device.

Besides the vendor and profile specific use of the address model the following rules shall be applied for the Slot:

- The slots 0 to 0x7FFF are freely usable within the scope of the addressed AP. For API 0 slots are vendor specific. For other APs further definitions specified in the related profile shall be applied.
- The slots 0x8000 to 0xFFFF are reserved for future use and shall not be used regardless of the AP.

Besides the vendor and profile specific use of the address model the following rules shall be applied for the Subslot:

- Subslot 0x0000 is freely usable within the scope of the addressed AP, if ARProperties.PullModuleAlarmAllowed is used.
- Subslot 0x0000 is used to address the module within the scope of the Pull Alarm to signal pulling of a module, if ARProperties.PullModuleAlarmAllowed is not used.
- The subslots 0x0001 to 0x7FFF are freely usable within the scope of the addressed AP. For API 0 subslots are vendor specific. For other APIs further definitions specified in the related profile shall be applied.
- The subslots 0x8000 to 0x8FFF shall be used to address interface and port submodules. They shall only be used in the default API (zero).
- The subslots 0x9000 to 0x9FFF shall be used to address interface and port submodules on the right or above delivering endnode in the special case system redundancy only. They shall only be used in the default API (zero).
- The subslots 0xA000 to 0xFFFF are reserved for future use and shall not be used regardless of the AP.

Each AP shall be addressed by an APIs (x) with $0 < x \leq 0xFFFFFFFF$. The usage of additional APs shall be restricted to official profile definitions.

7.1.6.2 Application service element

An application service element (ASE), as defined in ISO/IEC 9545, is a set of application functions that provide a capability for the interworking of application processes for a specific purpose. ASEs provide a set of services for conveying requests and responses to and from application processes and their objects.

The application layer offers the following ASEs:

IO Data ASE

The IO Data ASE provides a set of services to convey IO data cyclically. These data always belong to those slots/subslots that have been configured in terms of the Context ASE. IO Data Objects contain a status for transmission. Optionally the IO Data ASE offers the possibility to share the Input Data of one IO device with other IO devices. IO Data may also be read and written by a IO supervisor acyclically.

Record Data ASE

The Record Data ASE provides a set of services to convey acyclically data. The application of the IO controller or IO supervisor requests each transmission individually. The Record Data ASE can be related to all APs of an IO device.

LogBook Data ASE

The LogBook Data ASE provides a set of services to read LogBook data. The application of the IO controller or IO supervisor requests each transmission individually. The LogBook Data ASE can be related to all APs of an IO device.

Diagnosis ASE

The Diagnosis ASE provides services for the IO controller or IO supervisor to read Diagnosis information from an IO device.

Alarm ASE

The Alarm ASE provides a set of services to convey alarms issued by the IO device or IO controller. The assigned IO controller or IO device acknowledges the alarm.

Context ASE

The Context ASE provides a set of services to

- convey device parameter and configuration according device description
- identify AR endpoints
- maintain AR and CR parameter (timeouts, modes, ..)
- and to establish or release an association between individual APs.

Isynchronous Mode Application ASE

The Isochronous Mode Application ASE provides a set of services to parameterize and synchronize isochronous application processes.

Application Relationship ASE

The Application Relationship ASE (AR ASE) provides a description model for the separate AR types. This includes their transfer characteristics as well as their current communication states.

Physical Device Management ASE

The Physical Device Management ASE provides a set of services to manage the start-up of the physical device. It includes the assignment of a Station Name and further IP address parameter.

NOTE The term physical device implies that there is only one instance of this ASE for a “real” automation device regardless of the number of instances of device types within the automation device. Anyway, it will be only one instance for most devices.

7.1.6.3 Application process objects

An application process object (APO) is a network representation of a specific aspect of an application process (AP). Each APO represents a set of information and processing capabilities of an AP that are accessible through services of the application layer. APOs are used to represent these capabilities to other APs in a system.

In order to permit an AP to communicate with an AP of another device, APOs have to be available. Application process objects (virtual objects) represent existing process objects (real objects), that an application process has made visible and accessible to the communication (see Figure 19).

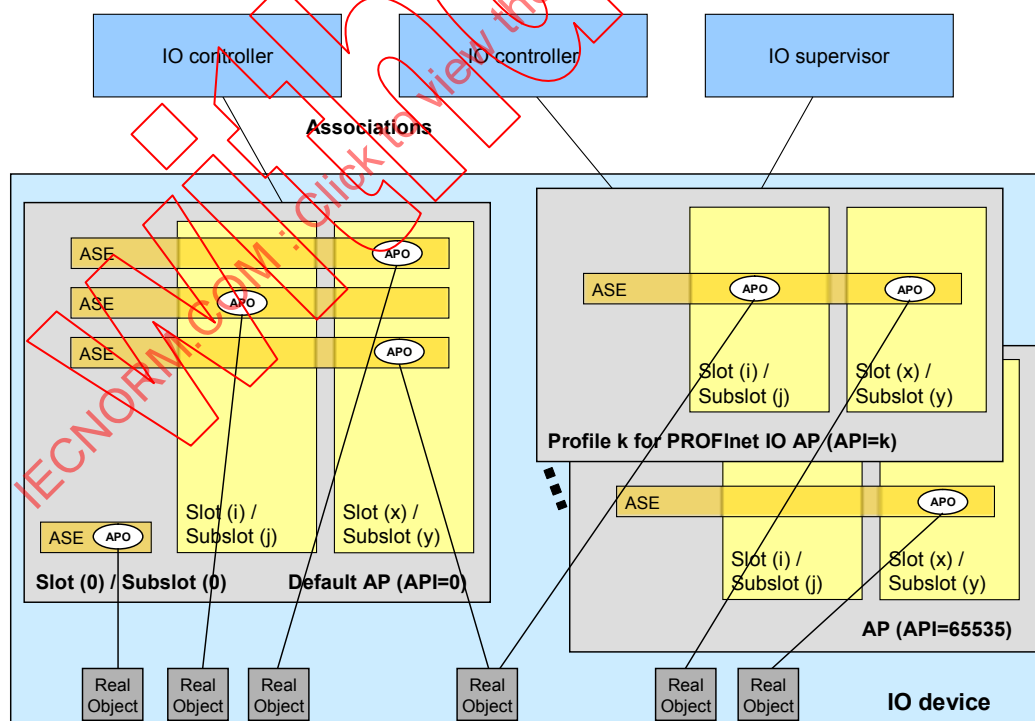


Figure 19 – Application Process with application process objects (APOs)

In Figure 20, a remote AP acting as Client may access the real object by sending requests through the APO that represents the real object. Local aspects of the AP convert between the network view (the APO) of the real object and the internal AP view of the real object.

Within the AP an APO is identified by slot, subslot and index. The address space which is defined by slot, subslot and index can be used by several APs.

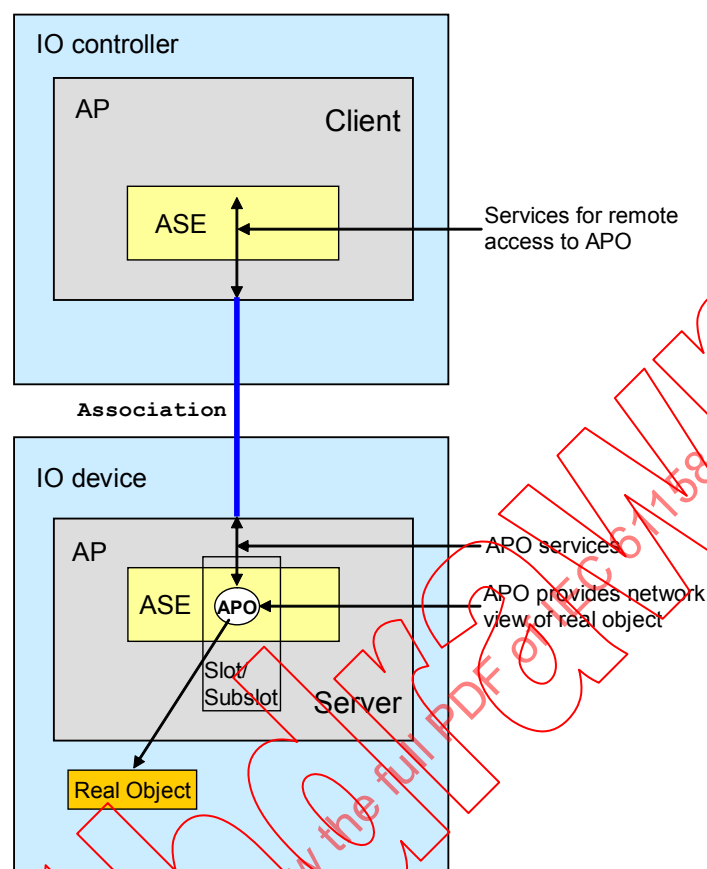


Figure 20 – Access to a remote APO

In Figure 21, a Client/Server and a Provider/Consumer association is shown. The Client may access the real object by sending requests through the APO that represents the real object. The Consumer may subscribe to the whole or to a part of the remote APO. The Provider/Consumer association is always related to the IO Data ASE. Local aspects of the AP convert between the network view (the APO) of the real object and the internal AP view of the real object.

NOTE The mapping of real data to network visible APOs is a local matter. However, the device vendor has to provide local means to avoid conflicts with overlapping outputs and writeable variables.

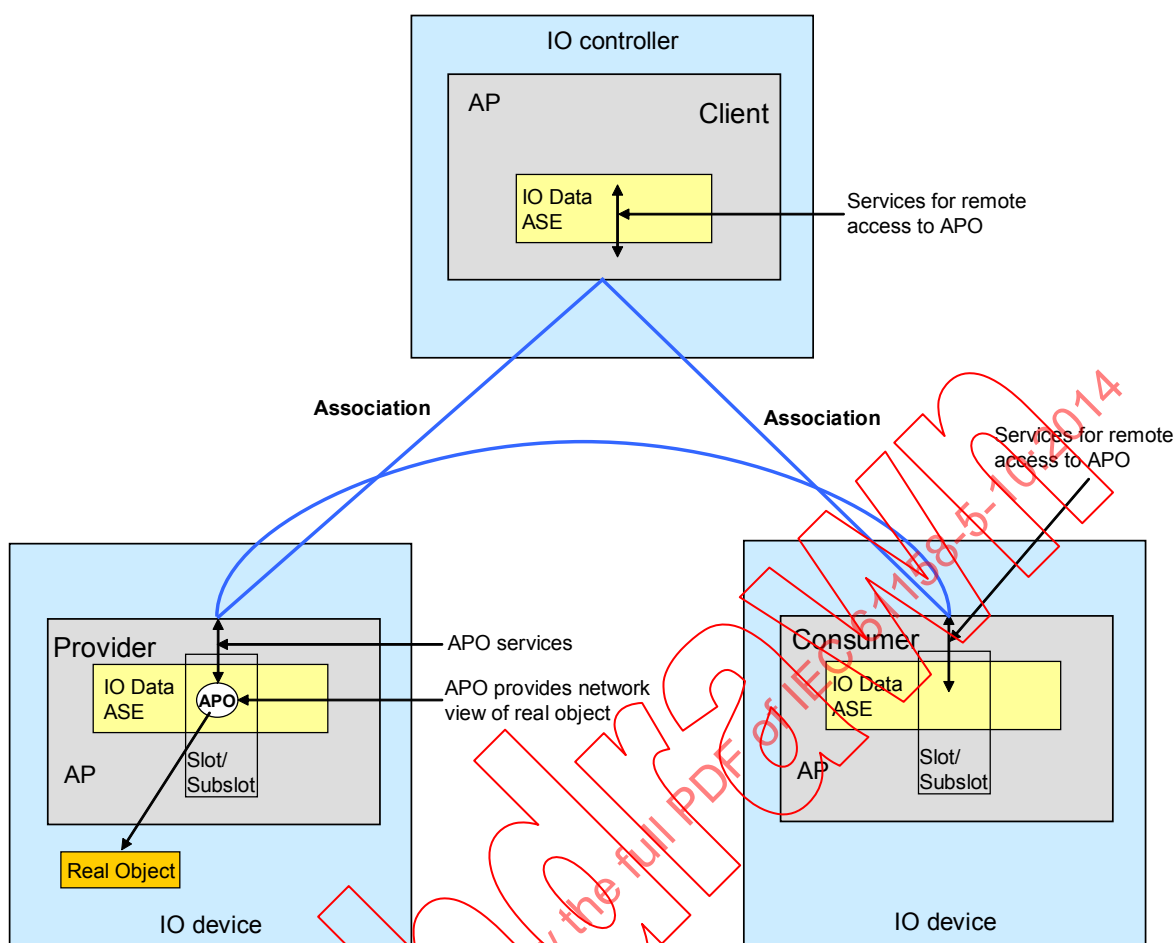


Figure 21 – Access to a remote APO for provider/consumer association

7.1.7 Application relationship

7.1.7.1 General

An Application Relationship (AR) is a co-operative relationship between two or more APs for the purpose of exchange of information and co-ordination of their joint operation (see Figure 22). This relationship is activated by the exchange of Application Protocol Data Units (APDU). The application layer uses different types of ARs which distinguish in their conveyance characteristic.

7.1.7.2 Application relationship endpoint

An AP has access to the communication using application relationship endpoints (AREP, see ISO 7498-1). One or more AREPs are fixed and uniquely assigned to an AP. These endpoints are addressed by the AP through an identifier (AREP ID). These identifiers are device specific and not defined by the communication itself. ARs are defined as a set of co-operating AREPs. Between two APs one or more ARs may exist, each one having unique AREPs. Figure 22 shows an example of one AR with two AREPs.

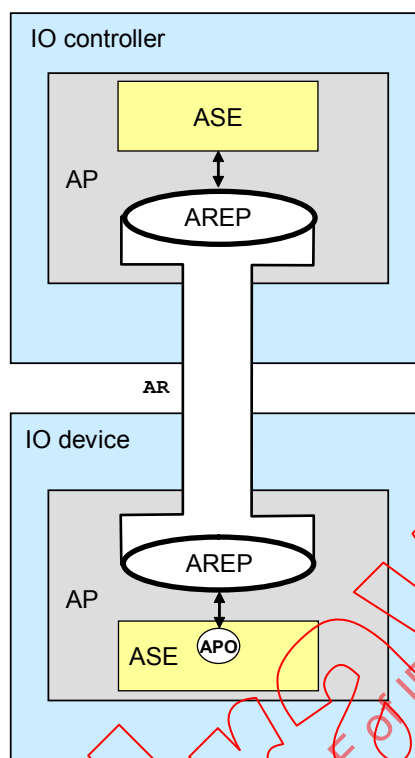


Figure 22 – Example of one AR with two AREPs

7.1.7.3 Overview of application relationships

The application layer offers the following AR types.

NOTE The AR can be identified with a unique UUID referred to as AR UUID.

IO AR: application relationship between the application process of one IO controller and the related application process of an IO device for the following purposes:

- cyclic exchange of the Input data with the IO controller by means of Input CR (BBUU CR)
- cyclic exchange of the Output data with the IO controller by means of Output CR (BBUU CR)
- cyclic exchange (multicast) of the Input data or Output data by means of M CR (BBUU-OM CR)
- acyclic data transfer for e.g. parameterization, configuration, IO data and diagnosis (Record Data objects) by means of RDCM CR (QQCB-CO CR)
- transmission of alarm data in both directions by means of Alarm CR (QQCB-CO CR)

The IO AR shall only be used together with API 0 to 0xFFFFFFFF.

Supervisor AR: application relationship between:

- the application processes of one IO supervisor and one IO device,
- the application processes of one IO parameter server and one related IO device,
- the application processes of one IO controller and one IO device,

for the following purposes:

- same functions as provided by IO AR with API 0 to 0xFFFFFFFF,

- additional takeover of an IO AR (controlled by supervisor)

Implicit AR: (AR UUID = UUID_NIL) application relationship between:

- the application processes of IO controller, IO devices, IO supervisor to Read ASE object values

The Implicit AR can be used together with all APIs.

7.1.8 FAL ASEs

Figure 23 shows the decentralized periphery FAL ASEs according to their OSI layer, including the common services FAL ASEs.

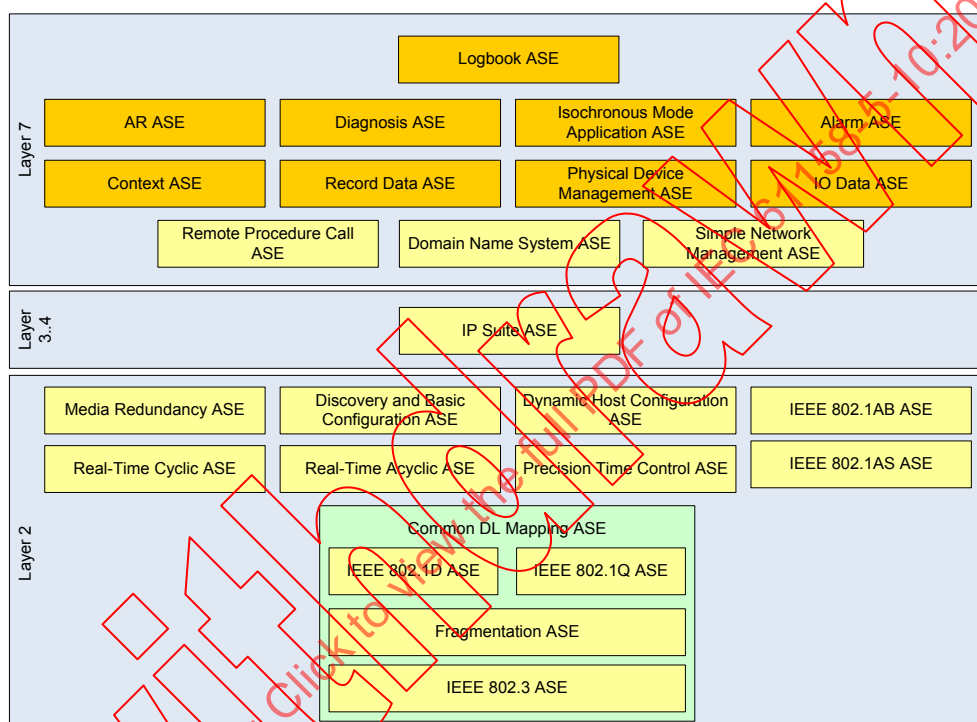


Figure 23 – FAL ASEs communication architecture

7.2 ASE data types

The data types defined in Clause 5 are supported.

7.3 ASEs

7.3.1 Record data ASE

7.3.1.1 Overview

In the Application Layer environment, application processes contain data that remote applications are able to read and write. The Record Data ASE defines attributes of general purpose Record Data objects and provides a set of services used to read, and write their values. Furthermore, the client may cancel a requested outstanding service due to special application conditions such as dynamic reconfiguration.

A particular Record Data object can be read and written partially or entirely (if supported by the application). The local connection between a Record Data object and a real world or physical object is not in the scope of this definition.

NOTE A Record Data object may be connected to one real world object for read and write, or, it can be assigned to one real object for reading and for another one for writing the value.

The Record Data ASE allows addressing Record Data objects related to specific layers of the addressing model as described in 7.1.6.1. The scope of a particular record data object is derived from the Index as described in 7.1.6.1.

Record Data objects belonging to

- a particular subslot within an AP are addressed by API, Slot Number, Subslot Number, and Index.
- a particular slot within an AP are addressed by API, Slot Number, and Index.
- a particular AP are addressed by API.
- a particular application relation are addressed by ARUUID and Index or if using a implicit AR by Target AR UUID and Index
- a device are addressed solely by Index.

Both IO controller and IO supervisor are able to read or write the value of some Record Data object from an IO device. Furthermore, an IO device is able to read or write the value of a Record Data object from an IO parameter server. An IO supervisor is able to read or write the value of a Record Data object from an IO controller. Using the attribute Device Access an IO supervisor is able to read or write all Record Data objects if permitted by the application process of the IO device.

The client application process may use the Context ASE to establish an association to get access to the Record Data objects of a server. Read access is also possible without a connection via the Implicit AR.

The access to Record Data objects is performed according to the Client/Server access model. The Client/Server model is characterized by a Client application sending a read or write request to a Server application that responds accordingly.

The formal model of the Record Data ASE is presented by the Record Data class specification, containing a description of its attributes, services as well as invocations, followed by a detailed service specification.

7.3.1.2 Binding of the Record Data services

The Record Data ASE's services are bound to lower level services as following, shown in Table 107:

Table 106 – Binding Record Data services

Service	Binding	Usage
Read	bound to RPC::Read	used by IO controller or IO supervisor
Read Query	bound to RPC::Read	used by IO controller or IO supervisor
Write	bound to RPC::Write	used by IO controller or IO supervisor

7.3.1.3 Record data class specification

7.3.1.3.1 Template

A Record Data object may be related to real object as shown in Figure 24.

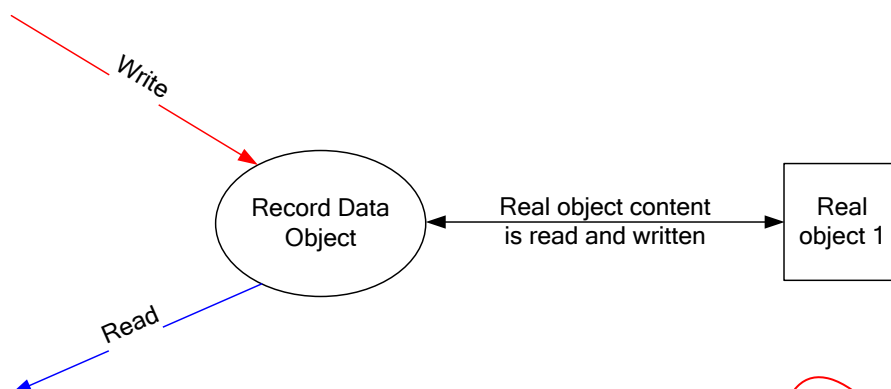


Figure 24 – Relation of a record data object to one real object

A Record Data object may be related to two or more real objects as shown in Figure 25.

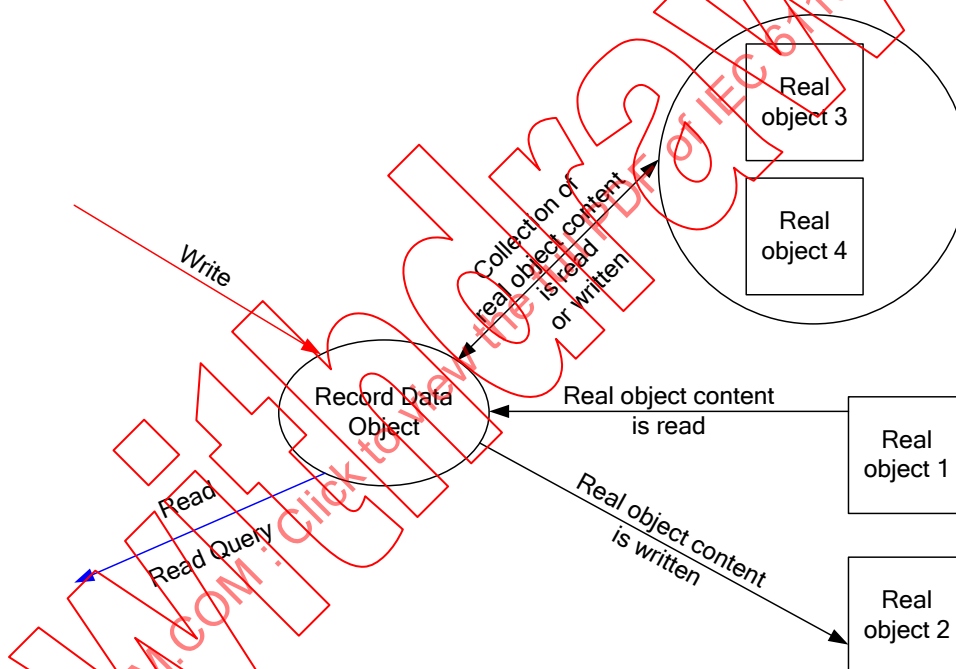


Figure 25 – Relation of a record data object to two real objects

A Record Data object is described by the following template:

ASE:	Record Data ASE
CLASS:	Record Data
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Identifier
1.1	(m) Attribute: API
1.2	(m) Attribute: Slot Number
1.3	(m) Attribute: Subslot Number
1.4	(m) Attribute: Index
2	(m) Attribute: Read Record Data Description
3	(m) Attribute: Read Partial Access

4	(m) Attribute:	Write Record Data Description
5	(m) Attribute:	Write Partial Access
6	(m) Attribute:	Shared Write Access
7	(o) Attribute:	Write Persistence Flag

SERVICES:

1	(m) OpsService:	Read
2	(o) OpsService:	Read Query
3	(m) OpsService:	Write
4	(m) OpsService:	Write Combined Object Container
5	(m) OpsService:	Local Write Multiple
6	(m) OpsService:	Local New Write Multiple

7.3.1.3.2 Attributes

Identifier

This key attribute is composed of attributes defining the context to which the Record Data object belongs. This Identifier is unique in the IO device and shall not be used by another object.

API

This attribute defines to which application process the Record Data object belongs.

Attribute type: Unsigned32

Allowed values: 0 as default application process. The values 1 to 0xFFFFFFFF are reserved for future use in profile guidelines.

Slot Number

This attribute defines to which module the Record Data object belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Subslot Number

This attribute defines to which subslot the Record Data object belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

Index

This attribute defines the index of the Record Data object.

Attribute type: Unsigned16

Allowed values: 0 to 0xFFFF

Read Record Data Description

This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-1 is used.

Read Partial Access

This attribute defines whether a partial read access to the Record Data object is supported or not. This attribute controls the behavior of the Read service. Partial access equal TRUE means that the content of the Record Data object may be read from the first octet up to the octet defined by the length given in the service request even if the length is shorter than the length of the Record Data object. The entire object will be read if the length given in the service is greater or equal than the length of the object in any case. Partial access equal FALSE means that the content of the Record Data object can only be read entirely. That implies that the length given in the service shall be equal or greater to the length of the object. Otherwise it shall return a negative response with the error code "Invalid type".

Attribute type: Boolean

Write Record Data Description

This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-1 is used.

Write Partial Access

This attribute defines whether a partial write access to the Record Data object is supported or not. This attribute controls the behavior of the Write service. Partial access equal TRUE means that the content of the Record Data object shall be written from the first octet up to the octet defined by the length given in the service request even if the length is shorter than the length of the Record Data object. The entire object will be written if the length given in the service is equal to the length of the object in any case. Partial access equal FALSE means that the content of the Record Data object can only be written entirely. That implies that the length given in the service shall be equal to the length of the object. Otherwise it shall return a negative response with the error code "Write Length Error". If the length in the Write service exceeds the length of the Record Data object a negative response with the error code "Write Length Error" shall be returned.

Attribute type: Boolean

Shared Write Access

This attribute defines whether a write access to the Record Data object is supported for client connected via a shared application relationship. This attribute controls the behavior of the Write service. Shared Write Access equal TRUE means that the content of the Record Data object can be written from client connected via a shared application relationship. In this case, the local application is responsible for data consistency. Shared Write Access equal FALSE means that the content of the Record Data object shall not be written from client connected via a shared application relationship.

Attribute type: Boolean

Write Persistence Flag

This optional attribute shall define the persistence behavior for the value of the Record Data Object during a power cycle. It shall be stored non-volatile if the value is set to TRUE. Otherwise the written value is volatile and after a power cycle missing.

NOTE 1 This attribute is a local device property implemented on behalf of the device vendor.

The application process shall implement the persistence storage behavior of Table 107 according to this attribute and the Write indication service parameter Prm Flag.

Table 107 – Persistence behavior for record data objects

Write Persistence Flag	Write.ind (Prm Flag)	Persistence Storage
FALSE	FALSE	shall be stored volatile
FALSE	TRUE	shall be stored volatile
TRUE	FALSE	shall be stored non-volatile before Write.rsp(+)
TRUE	TRUE	Write.ind: Shall be only intermediately stored volatile and shall immediately respond Write.rsp(+) before receiving Prm End indication Application Ready.req: Before issuing the Application Ready request, all intermediate stored data shall be checked for consistency, only consistent data set shall be stored non-volatile, shall be checked again for consistency in non-volatile memory, and send Application Ready request afterwards

NOTE Only consistent sets of data within the parameterization phase are stored non-volatile. An AR failure during this phase does not affect previously stored record data and the intermediate buffer is deleted.

NOTE 2 An application process may avoid persistence saving of unchanged values by local means, e.g. using checksums.

Attribute type: Boolean

7.3.1.3.3 Invocation of the Record Data object

For the invocation of the Record Data object the following rules apply:

- Record Data objects shall not exceed the total length of $2^{32}-65$ octets. The attribute Record Data description has to be set accordingly.
- At least one service shall be allowed for the access of the object.
- At least one AR shall have access to the object.
- The access rights have to be set accordingly to the allowed service(s).
- A record data service request (i.e. read, read query, write, or write multiple) shall be answered within less than 300 s; otherwise the client of the service may assume that the service timed out.

7.3.1.3.4 Focus of a record data object

The information of a record data object is filtered according to the following rules:

- If the record data object is addressed over the established AR or over implicit read with Target AR UUID, the information
 - is filtered according to the **expected** identification data (see Context ASE), if the addressed object is part of the addressed application relation.

NOTE 1 e.g. by getting diagnosis information from a slot being part of the expected identification data, only diagnosis related to submodules within the addressed slots, which are part of the expected identification data is returned.
 - is filtered according to the **real** identification data (see Context ASE) if the addressed object is not part of the addressed application relation.

NOTE 2 e.g. by getting diagnosis information from a slot not being part of the expected identification data, diagnosis related to **all** submodules within the addressed slots, which are part of the real identification data is returned.
- If the record data object is addressed over the implicit AR without specifying a Target AR UUID, the information is always filtered according to the **real** identification data (see Context ASE).
- If the record data object is on device layer, the information is always filtered according to the **real** identification data (see Context ASE).

The reason for these filter rules is that the user wants to see only the objects he is interested in.

7.3.1.3.5 Error code hierarchy of the Record Data object

The PNIO Status returned from the record data services shall be set according to the current error and shall consist of the elements Error code, Error Decode, Error code 1, and Error code 2.

If PNIO Status is set to (0x00, 0x00, 0x00, 0x00) the special case “No error” applies.

If an error shall be returned, Error code is set to IODReadRes when using the services Read or Read Query, else IODWriteRes when using the service Write.

A protocol error is indicated via Error decode = PNIO; for details regarding Error code 1 and Error code 2 see IEC 61158-6-10.

An application error on accessing a Record Data object is indicated via Error decode = PNORW. In this case the following error codes generation hierarchy applies for Error code 1. Error code 2 may be set to any user specific value.

- a) If a record data object is read and a target AR UUID has been specified (i.e. is not NIL) where the desired AR is not existing:
Error code 1 is set to “access / invalid parameter”; meaning that the desired target AR is not existent.
- b) If a record data object is written when the AR state is set to BACKUP:
Error code 1 is set to “access / backup”; meaning that the writing of records in AR state BACKUP is not allowed.
- c) If a record data object is accessed (read or written) where the API is not supported or not valid:
Error code 1 is set to “access / invalid area”; meaning that the desired API is not existent.
- d) If a record data object is accessed (read or written) where the slot does not exist or is empty:
Error code 1 is set to “access / invalid slot”; meaning that the desired API exists, but the desired slot is not existent or empty.
- e) If a record data object is accessed (read or written) where the subslot does not exist or is empty:
Error code 1 is set to “access / invalid subslot”; meaning that both desired API and slot exist, but the desired subslot is not existent or empty.
- f) If a record data object is accessed (read or written) where the index is not supported:
Error code 1 is set to “access / invalid index”; meaning that desired API, slot and subslot exist, but the desired index is not supported.
- g) If a record data object is written where write-access is not supported:
Error code 1 is set to “application / write error”, “access / type conflict”, “access / state conflict”, or “access / access denied”.
- h) If a record data object is read where read-access is not supported:
Error code 1 is set to “application / read error”, “access / type conflict”, “access / state conflict”, or “access / access denied”.
- i) Content errors in the structure of data record object to be written:
Error code 1 is set to “access / write length error”, “access / invalid range”, “access / invalid parameter”, or “access / invalid type”.
- j) Resource errors during the access onto the data record object:
Error code 1 is set to “resource / read constrain conflict”, “resource / write constrain conflict”, “resource / resource busy”, “resource / resource unavailable”, or “resource / user specific 12” to “resource / user specific 19”.
This indicates a temporary error and may lead to a successful access when the clients retries.
- k) Application errors during the access onto the data record object:
Error code 1 is set to “application / module failure”, “application / busy”, “application / version conflict”, “application / feature not supported”, or “application / user specific 1” to “application / user specific 6”.

7.3.1.4 Record data service specification

7.3.1.4.1 Read Services

7.3.1.4.1.1 Read

This confirmed service may be used to read the value of a Record Data object. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR. In addition to

this common read service, additional services are defined for read specific application data. All this read services use the same parameters, only the data field is different. Table 108 shows the parameters of the services. This service can also be used to read additional user and profile specific data.

The attribute Read Partial Access controls the behavior of the response in case the requested length is shorter than the length of the Record Data object (see attribute description).

Table 108 – Read

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
API	M	M(=)		
Target AR UUID	U	U(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Index	M	M(=)		
Seq Number	M	M(=)		
Length	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Length			M	M(=)
Data			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

API

If the scope of the record data object is subslot, slot or API, this parameter shall be used to address the desired AP.

Target AR UUID

If the scope of the record data object is application relation, this parameter shall be used in conjunction with the implicit AR to address the desired AR within the IO device.

Slot Number

If the scope of the record data object is subslot or slot, this parameter shall be used to address the desired slot (typically a module).

Subslot Number

If the scope of the record data object is subslot, this parameter shall be used to address the desired subslot (typically a submodule).

Index

The parameter Index shall be used to address the desired Record Data object. It is additionally used to derive the scope of the Record Data object.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16} - 1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The parameter Seq Number shall be incremented for each service request during a session.

NOTE 1 If for example a write service request had the Seq Number set to "17", a following read service request shall have the Seq Number set to "18".

When starting a new session the sequence number shall start with the last value of the session before. A confirmation with a Seq Number that is not outstanding will be ignored. An indication with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

Length

The parameter Length indicates the number of octets of a Input Data object that has to be read. The allowed length is from 2^0 to $2^{32} - 256$. If the requested length exceeds the actual length of the Input Data object element then the actual length with data shall be responded. If the requested length is shorter than the length of the Input Data object element then a parameter error shall be the result.

Result(+)

This parameter indicates that the service request succeeded.

Length

The parameter Length indicates the number of octets of the Data object that has been read.

Data

The parameter Data contains the value of the specific object which has been read and consists of the number of octets indicated in the Length. This parameter has to be composed of the data types defined in Clause 5. A specific meaning for the different services of this data exists, which is described in the special service parts. Table 109 shows the list of special read services.

Table 109 – Read Services

Service Name	Chapter
Read Output Data	7.3.2.5.16
Read Output Substitute Data	7.3.2.5.17
Read Input Data	7.3.2.5.9
Read LogBook Data	7.3.3.3.1
Read Diagnosis in channel coding for one subslot	7.3.4.3.5
Read Diagnosis in all codings for one subslot	7.3.4.3.6
Read Maintenance required in channel coding for one subslot	7.3.4.3.7
Read Maintenance required in all codings for one subslot	7.3.4.3.8
Read Maintenance demanded in channel coding for one subslot	7.3.4.3.9
Read Maintenance demanded in all codings for one subslot	7.3.4.3.10
Read Diagnosis, Maintenance, Qualified and Status for one subslot	7.3.4.3.11

Service Name	Chapter
Read Diagnosis in channel coding for one slot	7.3.4.3.12
Read Diagnosis in all codings for one slot	7.3.4.3.13
Read Maintenance required in channel coding for one slot	7.3.4.3.14
Read Maintenance required in all codings for one slot	7.3.4.3.15
Read Maintenance demanded in channel coding for one slot	7.3.4.3.16
Read Maintenance demanded in all codings for one slot	7.3.4.3.17
Read Diagnosis, Maintenance, Qualified and Status for one slot	7.3.4.3.18
Read Diagnosis in channel coding for one AR	7.3.4.3.19
Read Diagnosis in all codings for one AR	7.3.4.3.20
Read Maintenance required in channel coding for one AR	7.3.4.3.21
Read Maintenance required in all codings for one AR	7.3.4.3.22
Read Maintenance demanded in channel coding for one AR	7.3.4.3.23
Read Maintenance demanded in all codings for one AR	7.3.4.3.24
Read Diagnosis, Maintenance, Qualified and Status for one AR	7.3.4.3.25
Read Diagnosis in channel coding for one API	7.3.4.3.26
Read Diagnosis in all codings for one API	7.3.4.3.27
Read Maintenance required in channel coding for one API	7.3.4.3.28
Read Maintenance required in all codings for one API	7.3.4.3.29
Read Maintenance demanded in channel coding for one API	7.3.4.3.30
Read Maintenance demanded in all codings for one API	7.3.4.3.31
Read Diagnosis, Maintenance, Qualified and Status for one API	7.3.4.3.32
Read Diagnosis, Maintenance, Qualified and Status for one device	7.3.4.3.33
Read Module Diff Block	7.3.6.3.15
Read API Data	7.3.6.3.17
Read I&M0 Filter Data	7.3.6.3.18
Read I&M0 Data	7.3.6.3.19
Read I&M1 Data	7.3.6.3.21
Read I&M2 Data	7.3.6.3.23
Read I&M3 Data	7.3.6.3.25
Read I&M4 Data	7.3.6.3.27
Read Autoconfiguration Data	7.3.6.3.28
Read Expected Identification	7.3.6.3.8
Read Real Identification	7.3.6.3.10
Read Isochronous Mode Data	7.3.7.3.2
Read PDev Data	7.3.8.3.1
Read PD Real Data	7.3.8.3.2
Read PD Expected Data	7.3.8.3.3
Read PD Interface Data Real	7.3.8.3.4
Read PD Interface Adjust	7.3.8.3.6
Read PD Interface MRP Data Real	7.3.8.3.7
Read PD Interface MRP Data Check	7.3.8.3.18
Read PD Interface MRP Data Adjust	7.3.8.3.20
Read PD Interface FSU Data Adjust	7.3.8.3.22
Read PD NC Data Check	7.3.8.3.24

Service Name	Chapter
Read PD Sync Data	7.3.8.3.10
Read PD IR Data	7.3.8.3.8
Read PD IR Subframe Data	7.3.8.3.13
Read PD Port Data Real	7.3.8.3.25
Read PD Port Data Check	7.3.8.3.27
Read PD Port Data Adjust	7.3.8.3.29
Read PD Port FO Data Real	7.3.8.3.30
Read PD Port FO Data Check	7.3.8.3.32
Read PD Port FO Data Adjust	7.3.8.3.34
Read PD Port MRP Data Real	7.3.8.3.35
Read PD Port MRP Data Adjust	7.3.8.3.37
Read PD Port Statistic	7.3.8.3.38
Read PD Time Data	7.3.8.3.15
Read AR Data	7.3.9.5.1

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter selects one Error scheme for Error code 1 and 2; its coding is specified in IEC 61158-6-10.

Type: Unsigned8

Allowed Value: PNIORW

Error code 1

The Error code 1 assumes one of the following values:

read error, module failure, version conflict, feature not supported, user specific, invalid index, invalid slot/subslot, type conflict, invalid area, state conflict, access denied, invalid range, invalid parameter, invalid type, read constrain conflict, resource busy, resource unavailable, service cancelled.

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 2 Add Data 1 may be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 3 Add Data 2 may be used by device vendors to transmit specific error messages.

The negative response shall contain the values Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" if the addressed submodule has output data only.

7.3.1.4.1.2 Read Query

This optional confirmed service may be used to read the value of a Record Data object. To optimize the access to many small data objects a server may offer different combinations of this data behind one Index. In this case, the content of the Record Data object is defined by the Selector. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR. Table 110 shows the parameters of the service.

The attribute Read Partial Access controls the behavior of the response in case the requested length is shorter than the length of the Record Data object (see attribute description).

Table 110 – Read Query

Parameter name	req	ind	rsp	cnt
Argument	M	M(=)		
AREP	M	M(=)		
API	M	M(=)		
Target AR UUID	U	U(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Index	M	M(=)		
Seq Number	M	M(=)		
Length	M	M(=)		
Selector	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Length			M	M(=)
Data			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

API

If the scope of the record data object is subslot, slot or API, this parameter shall be used to address the desired AP.

Target AR UUID

If the scope of the record data object is application relation, this parameter shall be used in conjunction with the implicit AR to address the desired AR within the IO device.

Slot Number

If the scope of the record data object is subslot or slot, this parameter shall be used to address the desired slot (typically a module).

Subslot Number

If the scope of the record data object is subslot, this parameter shall be used to address the desired subslot (typically a submodule).

Index

The parameter Index shall be used to address the desired Record Data object. It is additionally used to derive the scope of the Record Data object.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16}-1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The parameter Seq Number shall be incremented for each service request during a session.

NOTE 1 If for example a write service request had the Seq Number set to "17", a following read service request shall have the Seq Number set to "18".

When starting a new session the sequence number shall start with the last value of the session before. A confirmation with a Seq Number that is not outstanding will be ignored. An indication with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

Length

The parameter Length indicates the number of octets of a Record Data object that has to be read. The allowed length is from 0 to $2^{32}-256$.

Selector

The parameter Selector is used by the server to identify the requested data. To optimize the access to many small data objects a server may offer different combinations of this data behind one Index and select the requested collection with the Selector.

Result(+)

This parameter indicates that the service request succeeded.

Length

The parameter Length indicates the number of octets of the Data object that has been read.

Data

The parameter Data contains the value of the object which has been read and consists of the number of octets indicated in the Length. This parameter has to be composed of the data types defined in Clause 5.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter selects one Error scheme for Error code 1 and 2; its coding is specified in IEC 61158-6-10.

Type: Unsigned8

Allowed Value: PNIORW

Error code 1

The Error code 1 assumes one of the following values:

read error, module failure, version conflict, feature not supported, user specific, invalid index, invalid slot/subslot, type conflict, invalid area, state conflict, access denied, invalid range, invalid parameter, invalid type, read constrain conflict, resource busy, resource unavailable, service cancelled.

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 2 Add Data 1 may be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 3 Add Data 2 may be used by device vendors to transmit specific error messages.

The negative response shall contain the values Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" if the addressed submodule has output data only.

7.3.1.4.2 Write Services

7.3.1.4.2.1 Write

This confirmed service may be used to write the value of a Record Data object. This service shall only be used in conjunction with the IO AR or Supervisor AR. In addition to this common write service, additional services are defined for write specific application data. All this write services use the same parameters, only the data field is different. Table 111 shows the parameters of the services. This service can also be used to write additional user and profile specific data.

The attribute Write Partial Access controls the behavior of the response in case the requested length is shorter than the length of the Record Data object (see attribute description).

Table 111 – Write

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Index	M	M(=)		
Seq Number	M	M(=)		

Parameter name	req	ind	rsp	cnf
Length	M	M(=)		
Data	M	M(=)		
Prm Flag		M		
Result(+)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

API

If the scope of the record data object is subslot, slot or API, this parameter shall be used to address the desired AR.

Slot Number

If the scope of the record data object is subslot or slot, this parameter shall be used to address the desired slot (typically a module).

Subslot Number

If the scope of the record data object is subslot, this parameter shall be used to address the desired subslot (typically a submodule).

Index

The parameter Index shall be used to address the desired Record Data object. It is additionally used to derive the scope of the Record Data object.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16} - 1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The parameter Seq Number shall be incremented for each service request during a session.

NOTE 1 If for example a write service request had the Seq Number set to "17", a following read service request shall have the Seq Number set to "18".

When starting a new session the sequence number shall start with the last value of the session before. A confirmation with a Seq Number that is not outstanding will be ignored. An indication with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

Length

The parameter Length indicates the number of octets of a Record Data object which has to be written. The allowed length is from 0 to $2^{32}-256$.

Data

The parameter Data contains the value of the Record Data object which has to be written and consists of the number of octets indicated in the Length of the request. This parameter has to be composed of the data types defined in Clause 5. A specific meaning for the different services of this data exists, which is described in the special service parts. Table 112 shows the list of special write services.

Table 112 – Write Services

Service Name	Chapter
Write Substitute Value	7.3.2.5.18
Write I&M1 Data	7.3.6.3.20
Write I&M2 Data	7.3.6.3.22
Write I&M3 Data	7.3.6.3.24
Write I&M4 Data	7.3.6.3.26
Write Isochronous Mode Data	7.3.7.3.1
Write PD Interface Adjust	7.3.8.3.5
Write PD IR Data	7.3.8.3.7
Write PD IR Subframe Data	7.3.8.3.12
Write PD Sync Data	7.3.8.3.9
Write PD Interface MRP Data Check	7.3.8.3.17
Write PD Interface MRP Data Adjust	7.3.8.3.19
Write PD Interface FSU Data Adjust	7.3.8.3.21
Write PD NC Data Check	7.3.8.3.23
Write MRP Port Data Adjust	7.3.8.3.36
Write PD Port Data Check	7.3.8.3.26
Write PD Port Data Adjust	7.3.8.3.28
Write PD Port FO Data Check	7.3.8.3.32
Write PD Port FO Data Adjust	7.3.8.3.33
Write PD Port MRP Data Adjust	7.3.8.3.36
Write PD Time Data	7.3.8.3.14

Prm Flag

The local indication parameter shall contain the value TRUE while parameterization during the connection establishment phase. Otherwise, the value shall be set to FALSE.

Attribute type: Boolean

NOTE 2 This parameter is used to control the persistent storage of data. It is utilized in fast start-up procedure.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter selects one Error scheme for Error code 1 and 2; its coding is specified in IEC 61158-6-10.

Type: Unsigned8

Allowed Value: PNIORW

Error code 1

The Error code 1 assumes one of the following values:

read error, module failure, version conflict, feature not supported, user specific, invalid index, invalid slot/subslot, type conflict, invalid area, state conflict, access denied, invalid range, invalid parameter, invalid type, read constrain conflict, resource busy, resource unavailable, service cancelled.

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 3 Add Data 1 may be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 4 Add Data 2 may be used by device vendors to transmit specific error messages.

The negative response shall contain the values Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" if the addressed submodule has output data only.

7.3.1.4.2.2 Write Combined Object Container

This confirmed service may be used to write the Comined Object Container to the combined object access point (see 7.1.4.5.6 for details). This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). Combined object Container belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the combined object container's data record index as specified in IEC 61158-6-10.

The parameter Data has a service specific structure and meaning. Table 113 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed submodule, which shall be the access point of the combined object. The submodules addressed by API, Slot Number and Subslot Number of a particular container entry shall be part of the combined object. The index shall be supported by the submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 113 – Data elements of Write Combined Object Container

Parameter name	service specific elements	req	ind
Data	List of Container	M	M(=)
	API	M	M(=)
	Slot	M	M(=)
	Subslot	M	M(=)
	Index	M	M(=)
	Data	M	M(=)

Data

This parameter contains the value of the object to write.

List of Container

The service transmits a number of data records with this list to write to the access point. Each list element consists of the following elements:

API

If the scope of the record data object is subslot, slot or API, this parameter shall be used to address the desired AP.

Slot Number

If the scope of the record data object is subslot or slot, this parameter shall be used to address the desired slot (typically a module).

Subslot Number

If the scope of the record data object is subslot, this parameter shall be used to address the desired subslot (typically a submodule).

Index

The parameter Index shall be used to address the desired Record Data object. It is additionally used to derive the scope of the Record Data object.

Length

The parameter Length indicates the number of octets of a Record Data object which has to be written. The allowed length is from 0 to $2^{32}-256$.

Data

The parameter Data contains the value of the Record Data object which has to be written and consists of the number of octets indicated in the Length of the request. This parameter has to be composed of the data types defined in Clause 5. A specific meaning for the different services of this data exists, which is described in the special service parts. Table 112 shows the list of special write services.

7.3.1.4.2.3 Local Write Multiple

This local service may be used to convey multiple Record Data objects within one APDU. This service shall only be used in conjunction with the IO AR or Supervisor AR. Table 114 shows the parameters of the service.

NOTE 1 It is possible to convey Record Data objects and attributes from multiple ASEs (e.g. Physical Device Management ASE) in one APDU.

This local service is mapped by building a surrounding data record from the individual address information (API, slot, subslot, index), sequence number and data of the particular data records. This data record is then transmitted to the device by using the service Write.req (see 7.3.1.4.2.1).

Within the device such a surrounding data record is recognized by its special index (see IEC 61158-6-10 for its definition; in particular the APDU substitution IODWriteMultipleReq). Within the device a receipt of such a data record (indicated via Write.ind) is then mapped to the local service Local New Write Multiple.

Table 114 – Local Write Multiple

Parameter name	req	cnf
Argument	M	
AREP	M	
Seq Number	M	
List of Data Records	M	
API	M	
Slot Number	M	
Subslot Number	M	
Index	M	
Seq Number	M	
Length	M	
Data	M	
Prm Flag		
Result(+)		S
AREP		M
Seq Number		M
List of Results		M
Seq Number		M
PNIO Status		M
Add Data 1		M
Add Data 2		M
Result(-)		S
AREP		M
Seq Number		M
Error Decode		M
Error code 1		M
Error code 2		M
Add Data 1		M
Add Data 2		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16}-1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The value parameter Seq Number shall be incremented for each service request during a session. Starting a new session the sequence number shall start with the last value of the session before. Confirmations with a Seq Number that is not outstanding will be ignored. Indications with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

NOTE 2 This seq number is used for the write multiple service.

List of Data Records

The service transmits a number of data records with this list. Each list element is contained of the following elements:

API

This parameter shall be used to address the desired API.

Slot Number

The parameter Slot Number is used in the destination device for addressing the desired Record Data object in a specific slot (typically a module).

Subslot Number

The parameter Subslot Number is used in the destination device for addressing the desired Record Data object in a specific sub-element (typically a submodule).

Index

The parameter Index is used in the destination device for addressing the desired Record Data object.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16}-1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The value parameter Seq Number shall be incremented for each service request during a session. Starting a new session the sequence number shall start with the last value of the session before. Confirmations with a Seq Number that is not outstanding will be ignored. Indications with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

NOTE 3 This seq number is used for the individual data record.

Length

The parameter Length indicates the number of octets of a Record Data object which has to be written. The allowed length is from 0 to $2^{32}-65$.

Data

The parameter Data contains the value of the Record Data object which has to be written and consists of the number of octets indicated in the Length of the request. This parameter has to be composed of the data types defined in Clause 5.

Prm Flag

The local indication parameter shall contain the value TRUE while parameterization during the connection establishment phase. Otherwise, the value shall be set to FALSE.

Attribute type: Boolean

NOTE 4 This parameter is used to control the persistent storage of data. It is utilized in fast start-up procedure.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.1.4.2.4 Local New Write Multiple

This local service indicates the receipt of multiple Record Data objects within one APDU. This service shall only be used in conjunction with the IO AR or Supervisor AR. Table 115 shows the parameters of the service.

NOTE 1 It is possible to convey Record Data objects and attributes from multiple ASEs (e.g. Physical Device Management ASE) in one APDU.

This local service is mapped by receipt of a surrounding data record indicated by receiving the special index (see IEC 61158-6-10 for its definition; in particular the APDU substitution IODWriteMultipleReq) on the service indication Write.ind (see 7.3.1.4.2.1).

The individual address information (API, slot, subslot, index), sequence number and data of the particular data records are extracted from the surrounding data record by the application and presented in this service's indication.

If the surrounding data record is not decodable (due to its internal structure) a result(-) is returned.

Otherwise the particular data records are applied one by one onto the application by using the address information and the data. The results of the particular data record writes are collected together (see IEC 61158-6-10 for its definition; in particular the APDU substitution IODWriteMultipleReq) and returned via Write.rsp(+).

The local service Write Multiple.cnf then extracts the particular results from the returned information and returns it to the application.

Table 115 – Local New Write Multiple

Parameter name	ind	rsp
Argument	M(=)	
AREP	M(=)	
Seq Number	M(=)	
Slot Number	M(=)	
Subslot Number	M(=)	
Index	M(=)	
Seq Number	M(=)	
Length	M(=)	
Data	M(=)	
Prm Flag	M	
Result(+)		S
AREP		M
Seq Number		M
List of Results		M
Seq Number		M
PNIO Status		M
Add Data 1		M
Add Data 2		M
Result(-)		S
AREP		M
Seq Number		M
Error Decode		M
Error code 1		M
Error code 2		M
Add Data 1		M
Add Data 2		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16}-1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The value parameter Seq Number shall be incremented for each service request during a session. Starting a new session the sequence number shall start with the last value of the session before. Confirmations with a Seq Number that is not outstanding will be ignored. Indications with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

NOTE 2 This seq number is used for the write multiple service.

List of Data Records

The service transmits a number of data records with this list. Each list element is contained of the following elements:

API

This parameter shall be used to address the desired API.

Slot Number

The parameter Slot Number is used in the destination device for addressing the desired Record Data object in a specific slot (typically a module).

Subslot Number

The parameter Subslot Number is used in the destination device for addressing the desired Record Data object in a specific sub-element (typically a submodule).

Index

The parameter Index is used in the destination device for addressing the desired Record Data object.

Seq Number

The parameter Seq Number is used by the server to identify the duplication of services with a sequence number. The range of this service parameter is from 0 to $2^{16}-1$. The requesting application process shall provide a unique Seq Number to each outstanding service request. The value parameter Seq Number shall be incremented for each service request during a session. Starting a new session the sequence number shall start with the last value of the session before. Confirmations with a Seq Number that is not outstanding will be ignored. Indications with a Seq Number that is not outstanding will be rejected. The Seq Number shall be maintained for each established AR separately.

NOTE 3 This seq number is used for the individual data record.

Length

The parameter Length indicates the number of octets of a Record Data object which has to be written. The allowed length is from 0 to $2^{32}-65$.

Data

The parameter Data contains the value of the Record Data object which has to be written and consists of the number of octets indicated in the Length of the request. This parameter has to be composed of the data types defined in Clause 5.

Prm Flag

The local indication parameter shall contain the value TRUE while parameterization during the connection establishment phase. Otherwise, the value shall be set to FALSE.

Attribute type: Boolean

NOTE 4 This parameter is used to control the persistent storage of data. It is utilized in fast start-up procedure.

Result(+)

This parameter indicates that the service request itself succeeded; note however that the PNIO Status of some or even all write results may indicate an error.

List of Results

This list contains the result of each data record access. Each list element is contained of the following elements:

PNIO Status

This parameter indicates the error; see 7.3.1.3.5 for details.

Type: Unsigned32

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 5 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 6 Add Data 2 can be used by device vendors to transmit specific error messages.

Result(-)

This parameter indicates that the service request failed totally and none of the data records has been written.

PNIO Status

This parameter indicates the error; see 7.3.1.3.5 for details.

Type: Unsigned32

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 7 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 8 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.2 IO Data ASE

7.3.2.1 Overview

In the application layer environment, each application process of an IO device shall contain exactly one IO Data object for each application process instance. It is structured by means of slots and subslots. Subslots represent IO data that shall be cyclically conveyed over the

network. Therefore the IO Data ASE defines attributes of the IO Data objects and provides a set of services used to get and set the buffer to transport their values. The IO Data objects are composed of an Input Data object and an Output Data object.

An Input Data object or an Output Data object shall be partitioned in Input Data or Output Data Elements which represent the IO Data of the slot/subslot.

Additional services are provided to read acyclically the values of the IO Data objects and to indicate new values for the Input Data object and the Output Data object. Furthermore services are provided to indicate the presence and to get the content of new consumed Input Data.

The IO Data objects are implicitly addressed through the related services. The granularity of input or output data in a Server/Provider is according to the correspondent configuration attributes.

The I/O Data ASE uses the Client/Server and Provider/Consumer access model. The Client/Server model is characterized by a Client application conveying the value for the Output Data object to the Server's buffer. The Server application fetches this value by a Get Output service. The receipt of a new value is indicated by the New Output service. The Server application conveys the value of its Input Data object via the Set Input service to the Server's buffer as soon as one of the real object values changes. The Client application gets this value by the Get Input service. The transmission of the Server buffer to the Client buffer and vice versa is done asynchronously to the Get Output and Set Input services.

The formal model of the IO Data ASE is described next, followed by a description of its services. Furthermore, the IO Data ASE represents the real input and output structure of an IO device. The relations between real IO data structure and conveyance of IO data via ARs are described within the Context ASE.

Figure 26 shows the IO service interactions for the IO controller and IO device.

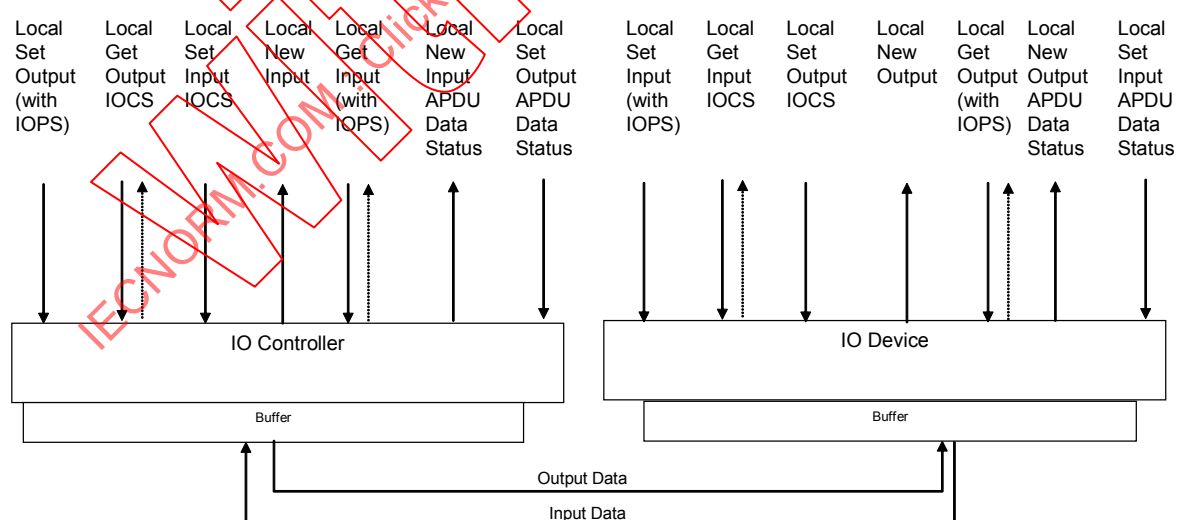


Figure 26 – Overview IO ASE service interactions

The services interact with local buffers maintained by the application layer protocol. This protocol provides mechanisms for cyclic exchange of buffers between the devices. Local services provide a means to get or set the values of IO data together with their status values. The status values are referred to as IOPS and IOCS.

7.3.2.2 IOPS / IOCS behavior

7.3.2.2.1 IOPS

The IOPS (Input Output Provider Status) describes the validness of data source as being seen by the subscriber of the data. It belongs to Output Data for an IO controller and to Input Data for an IO device.

NOTE If a submodule does not contain any input or output data it is considered as an Input Data item with the length of Input Data attribute set to zero.

The detecting instance shall set the IOPS accordingly and immediately. The IOPS is set by local means; e.g. on configuration errors, on submodule or module pulled, on insufficient parameterization state of the submodule (application ready pending), on diagnosis...

The IOPS may be set by individual instances participating when transmitting the input data: by the submodule (including the sensor), by the module, by the IO device, and by the IO controller (locally, e.g. data hold timeout).

The IOPS allows identifying the first instance which detected the invalid content, with the help of the following values:

- GOOD – data is valid
- BAD_BY_SUBSLOT – data is invalid as detected by the submodule
- BAD_BY_SLOT – data is invalid as detected by the module
- BAD_BY_DEVICE – data is invalid as detected by the IO device
- BAD_BY_CONTROLLER – data is invalid as detected by the IO controller

Only valid data with IOPS==GOOD shall be processed by the application. In all other cases (BAD_BY_XXX) the application shall raise an error locally (e.g. to the application program in the IO controller) or use default values (e.g. zero, last valid value, substitute value; on the IO device) according to its configuration and parameterization.

The IOPS shall be set independent of the corresponding IOCS; e.g. if the IOCS is set to BAD_BY_CONTROLLER but the IO device is still able to deliver valid data, the IOPS shall be set to GOOD.

7.3.2.2.2 IOCS

The IOCS (Input Output Consumer Status) indicates whether the application (controller or device) is processing the value. It belongs to Input Data for an IO controller and to Output Data for an IO device.

The IOCS is set by individual instances participating when transmitting the output data: by the submodule (including the actor), by the module, by the IO device, and by the IO controller (for an input module locally, e.g. data hold timeout; for an output module e.g. due to being in operating state Stop).

The detecting instance shall set the IOCS accordingly and immediately. The IOCS is set by local means; e.g. on configuration errors, on submodule or module pulled, on insufficient parameterization state of the submodule (application ready pending), on diagnosis...

The IOCS allows identifying the first instance which does no longer process the content, with the help of the following values:

- GOOD – data is processed by the application
- BAD_BY_SUBSLOT – data is not being processed by the application as detected by the submodule

- BAD_BY_SLOT – data is not being processed by the application as detected by the module
- BAD_BY_DEVICE – data is not being processed by the application as detected by the device
- BAD_BY_CONTROLLER – not being processed by the application as detected by the submodule.

For an IO device applies: The IOCS shall be set independent of the corresponding IOPS; i.e. if the IOPS is set to BAD_BY_xxx but the IO device is still able to process the data, the IOCS shall be kept on GOOD.

For an IO controller applies: The IOCS may be set dependent or independent of the corresponding IOPS. The possible action of the IO controller is beyond the scope of this service definition.

NOTE The best practice pattern for the handling within the IO controller is: If IO device is not able to process the output data (indicated by setting the IOCS to "BAD_BY_XXX"), the IO controller's application should set the corresponding IOPS to "BAD_BY_CONTROLLER". Thus allows the IO controller's application to apply initial values whenever the IO device is able to process the output data again; indicated by setting the IOCS to "GOOD" and sending a "return of submodule" alarm. During process of the alarm within the IO controller's application the initial value is applied to the output data and in consequence the IOPS is set to "GOOD" again.

The possible action of the IO device is beyond the scope of this service definition.

7.3.2.2.3 Behavior on Connect, Plug, Prm Begin and Application Ready

If the AR Info responded in the Module Diff Block for a particular submodule states APPLICATION_READY_PENDING, the corresponding IOPS/IOCS shall be set to BAD_BY_SUBSLOT.

The IOPS/IOCS shall be set to GOOD earliest after finishing the submodule parameterization and if no further circumstances (e.g. application) hinder the validness of the data.

If the AR Info responded in the Module Diff Block for a particular submodule states SUPERORDINATED_LOCKED, LOCKED_BY_IO_CONTROLLER, or LOCKED_BY_IO_SUPERVISOR, the corresponding IOPS/IOCS shall be set to BAD_BY_SUBSLOT.

The IOPS/IOCS shall be set to GOOD earliest if the application relation gets the ownership and if no further circumstances (e.g. application) hinder the validness of the data.

If the Ident Info responded in the Module Diff Block for a particular submodule states WRONG or NO, the corresponding IOPS/IOCS shall be set to BAD_BY_SUBSLOT.

The IOPS/IOCS shall be set to GOOD earliest after plugging the right submodule, if the application relation gets the ownership, and if no further circumstances (e.g. application) hinder the validness of the data.

If the Module State responded in the Module Diff Block for a particular module states WRONG_MODULE or NO_MODULE, all corresponding IOPS/IOCS of that module's submodules shall be set to BAD_BY_SLOT.

The IOPS/IOCS shall be set to GOOD earliest for a particular submodule after plugging the right module with that submodule, if the application relation gets the ownership, and if no further circumstances (e.g. application) hinder the validness of the data.

7.3.2.2.4 Behavior on transistions

If an IO device changes a particular IOPS/IOCS, it shall indicate a reason to the IO controller.

- On transition from GOOD to BAD_BY_xxx:
 - If the submodule's ownership changed to an IO supervisor or to the application (i.e. superordinated locked), the reason is indicated by a Controller by supervisor alarm.
 - If the submodule has been pulled, the reason is indicated by a Pull alarm.
 - If the application indicates the invalidness of the IO data, the reason is indicated e.g. by a coming diagnosis alarm, by setting ProviderState to "Stop"...
- On transition from BAD_BY_xxx to GOOD:
 - If the submodule's ownership changed back to the IO controller, the reason is indicated by Released alarm.
 - If the submodule has been plugged again, the reason is indicated by a Plug alarm.
- If the application indicates the validness of the IO data, the reason is indicated e.g. by a going diagnosis alarm, by setting ProviderState to "Run"... Additionally a Return of Submodule alarm shall be indicated.

7.3.2.3 APDU status behavior

Services are provided to set and indicate a change of the application data status (APDU Data Status).

7.3.2.4 IO Data class specification

7.3.2.4.1 Input Data class specification

7.3.2.4.1.1 Template

The Input Data object is described by the following template:

ASE:	IO Data ASE
CLASS:	Input Data
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: List of APs
2.1	(m) Attribute: API
2.2	(m) Attribute: List of Input Data Elements
2.2.1	(m) Attribute: Slot Number
2.2.2	(m) Attribute: List of Subslots
2.2.2.1	(m) Attribute: Subslot Number
2.2.2.2	(m) Attribute: Input Data Object Format
2.2.2.2.1	(m) Attribute: Length Input Data
2.2.2.2.2	(m) Attribute: Input Data Structure
2.2.2.3	(m) Attribute: Length IOPS
2.2.2.4	(m) Attribute: IOPS
2.2.2.5	(m) Attribute: Length IOCS
2.2.2.6	(m) Attribute: IOCS
SERVICES:	
1	(m) OpsService: Local Set Input
2	(m) OpsService: Local Set Input IOCS
3	(m) OpsService: Local Get Input
4	(m) OpsService: Local Get Input IOCS
5	(o) OpsService: Local New Input
6	(m) OpsService: Local Set Redundancy
7	(m) OpsService: Local Set State

- 8 (m) OpsService: Local Data State Changed
- 9 (m) OpsService: Read Record Input Data Object Element

7.3.2.4.1.2 Attributes

Implicit

The key attribute Implicit indicates that the Input Data object is implicitly addressed by the services.

List of APs

One AP is composed of the following list elements:

API

This attribute defines to which application process the Input Data Elements belongs.

Attribute type: Unsigned32

Allowed values: 0 as default application process. The values 1 to 0xFFFFFFFF are reserved for future use in profile guidelines.

List of Input Data Elements

One Input Data Element is composed of the following list elements:

Slot Number

This attribute defines to which module the Input Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the Input Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF are allowed

NOTE If ARProperties.PullModuleAlarmAllowed is not used, subslot number zero is used in conjunction with AlarmType "Pull" to address a certain module within the IO device. In this case, it shall not contain channels and IO data.

Input Data Object Format

This attribute consists of the following attributes.

Length Input Data

The attribute Length Input Data defines the number of octets of the Input Data Element without IOPS.

Attribute type: Unsigned16

Allowed values: 0 to 1 439

Input Data Structure

This attribute contains either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-1 is used.

The Input Data shall be transmitted consistent from the sending buffer to the receiving buffer completely. However, the application may get or set only parts of a single data object including the IOPS.

This field is part of the GSD.

Furthermore, if the Input Data are the source of a Multicast Provider CR then the associated Output Data of the Multicast Consumer CR shall have exactly the same data structure.

Length IOPS

The attribute Length IOPS defines the number of octets of the provider status.

Attribute type: Unsigned8

Allowed values: 1, (2 to 255 are reserved for future use)

IOPS

The attribute defines the provider status for the Input Data Object Element which belongs to the subslot of the IO device.

Attribute type: Unsigned8

Allowed values:

GOOD – content of Input Data Object Element is valid,

BAD_BY_SUBSLOT – content of Input Data Object Element is invalid; it was detected by the submodule,

BAD_BY_SLOT – content of Input Data Object Element is invalid; it was detected by the module,

BAD_BY_DEVICE – content of Input Data Object Element is invalid; it was detected by the IO device,

Special case: BAD_BY_CONTROLLER – content of Input Data Object Element is invalid, only locally detected by the IO controller itself e.g. through a communication timeout. This value shall only be used with the service parameter IOPS together with the service Get Input at the IO controller. It shall never be applied with the Set Input service at the IO device.

Length IOCS

The attribute Length IOCS defines the number of octets of the consumer status.

Attribute type: Unsigned8

Allowed values: 1, (2 to 255 are reserved for future use)

IOCS

The attribute defines the consumer status for the Input Data object element as an application feedback of the IO controller.

Attribute type: Unsigned8

Allowed values:

GOOD – the Input Data Object Element could be successfully processed by the application process of the IO controller,

BAD_BY_CONTROLLER – the Input Data Object Element could not be successfully processed by the application of the IO controller e.g. in case of an error or as cause of the operation state („stop“) of the IO controller

Special case: BAD_BY_DEVICE – the IO device has locally detected problems to convey data. It shall never be applied with the Set Input IOCS service at the IO controller.

7.3.2.4.1.3 Invocation of the Input Data object

For the invocation of the Input Data object the following rules apply:

- Input Data object elements data shall not exceed the total length of 1 439 octets. The attributes List of Input Data Elements, Input Data Object Format respectively Format and Length have to be set accordingly.

- The relation of an Input Data object element to a Input CR, Output CR (only IOCS), M CR will be handled in the Context ASE.
- The sum of all Input Data object elements related to one Input CR or M CR including IOPS and IOCS of the related Output Data object elements shall not exceed the total length of 1 440 octets.
- One Input Data object shall be invoked in an application process of an IO device.

7.3.2.4.2 Output data class specification

7.3.2.4.2.1 Template

The Output Data object is described by the following template:

ASE: IO Data ASE

CLASS: Output Data

CLASS ID: not used

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Key Attribute:	Implicit
2	(m) Attribute:	List of APs
2.1	(m) Attribute:	API
2.2	(m) Attribute:	List of Output Data Elements
2.2.1	(m) Attribute:	Slot Number
2.2.2	(m) Attribute:	List of Subslots
2.2.2.1	(m) Attribute:	Subslot Number
2.2.2.2	(m) Attribute:	Output Data Object Format
2.2.2.2.1	(m) Attribute:	Length Output Data
2.2.2.2.2	(m) Attribute:	Output Data Structure
2.2.2.3	(m) Attribute:	Length IOPS
2.2.2.4	(m) Attribute:	IOPS
2.2.2.5	(m) Attribute:	Length IOCS
2.2.2.6	(m) Attribute:	IOCS
2.2.2.7	(m) Attribute:	Safe State Behavior Changeable
2.2.2.8	(m) Attribute:	Substitute Mode
2.2.2.9	(m) Attribute:	Output Substitute Data Valid
2.2.2.10	(m) Attribute:	Output Substitute Data

SERVICES:

1	(m) OpsService:	Local Set Output
2	(m) OpsService:	Local Set Output IOCS
3	(m) OpsService:	Local Get Output
4	(m) OpsService:	Local Get Output IOCS
5	(o) OpsService:	Local New Output
6	(m) OpsService:	Local Set Provider State
7	(m) OpsService:	Read Record Output Data Object Element
8	(m) OpsService:	Read Substitute Value
9	(m) OpsService:	Write Substitute Value

7.3.2.4.2.2 Attributes

Implicit

The key attribute Implicit indicates that the Output Data object is implicitly addressed by the services.

List of APs

One AP is composed of the following list elements:

API

This attribute defines to which application process the Output Data Elements belongs.

Attribute type: Unsigned32

Allowed values: 0 as default application process; 1 to 0xFFFFFFFF are reserved for the use in profile guidelines.

List of Output Data Elements

One Output Data Element is composed of the following list elements:

Slot Number

This attribute defines to which module the Output Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the Output Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF are allowed

NOTE 1 Subslot Number 0 is used in conjunction with Alarm Type "Pull" to address a certain module within the IO device. It is never a "real" submodule and it does not contain Output Data Elements.

Output Data Object Format

This attribute consists of the following attributes.

Length Output Data

The attribute Length Output Data defines the number of octets of the Output Data Element without IOPS.

Attribute type: Unsigned16

Allowed values: 1 to 1 439

Output Data Structure

This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-1 is used.

The Output Data shall be transmitted consistent from the sending buffer to the receiving buffer completely. However, the application may get or set only parts of a single data object including the IOPS.

This field is part of the GSD.

Furthermore, if the Output Data are the sink of a Multicast Consumer CR then the associated Input Data of the Multicast Provider CR shall have exactly the same data structure.

Length IOPS

The attribute Length IOPS defines the number of octets of the provider status.

Attribute type: Unsigned8

Allowed values: 1, (2 to 255 are reserved for future use)

IOPS

The attribute defines the provider status for the Output Data Object Element which belongs to the subslot of the IO device.

Attribute type: Unsigned8

Allowed values:

GOOD - the Output Data Object Element was successfully processed by the application process of the IO controller,

BAD_BY_CONTROLLER - the Output Data Object Element could not be successfully processed by the application of the IO controller e.g. in case of an error or as cause of the operation state („stop“) of the IO controller

Special case: BAD_BY_DEVICE – the IO device has locally detected problems to convey data. It shall never be applied with the Set Input service at the IO controller.

Length IOCS

The attribute Length IOCS defines the number of octets of the consumer status.

Attribute type: Unsigned8

Allowed values: 1, (2 to 255 are reserved for future use)

IOCS

The attribute defines the consumer status for the Output Data Object Element which belongs to the subslot of the IO device.

Attribute type: Unsigned8

Allowed values:

GOOD – the Output Data Object Element was successfully processed,

BAD_BY_SUBSLOT - the Output Data Object Element was not successfully processed, it was detected by the submodule,

BAD_BY_SLOT - the Output Data Object Element was not successfully processed, it was detected by the module,

BAD_BY_DEVICE - the Output Data Object Element was not successfully processed, it was detected by the IO device,

Special case: BAD_BY_CONTROLLER - only locally detected by the IO controller itself e.g. through a communication timeout. This value shall only be used with the service parameter IOCS together with the service Get Output IOCS at the IO controller. It shall never be applied with the Set Output IOCS service at the IO device.

Safe State Behavior Changeable

The Safe State Behavior Changeable attribute defines if this behavior is adjustable.

Attribute type: Boolean

Substitution Mode

The Substitution Mode attribute defines the handling of the output data subslot in case of a failure. This behavior shall be applied if the IOPS value contains other values than GOOD.

Attribute type: Unsigned16

Allowed Values:

ZERO – is a submodule specific function to set all output values of the Output Data element to zero (IOPS==GOOD) or inactive (IOPS==BAD),

LAST_VALUE – all output values of the Output Data element shall be set to the last valid (IOPS==GOOD) values of the Get Output confirmation service primitive,

REPLACEMENT_VALUE – all output values of the Output Data element shall be set to the value of the attribute Safe State Value

PROFILE_SPECIFIC1 - PROFILE_SPECIFIC255 reserved for profile specification

Substitute Data Valid

The Substitute Data Valid attribute defines the validity of the attribute Substitute Data.

Attribute type: Boolean

Allowed Values:

TRUE shall be used in conjunction with Substitution Mode ZERO, LAST_VALUE, REPLACEMENT_VALUE and PROFILE_SPECIFIC to indicate valid data.

FALSE shall be used in conjunction with Substitution Mode ZERO and PROFILE_SPECIFIC to indicate invalid data. In this case the data content shall be set to zero.

Substitute Data

The Substitute Data attribute defines the replacement value of the output data for the subslot in case of a failure.

The attribute type shall be the same as defined in the Output Data Structure attribute.

The value of the Substitute Data attribute shall be the last valid value of the Output Data element in case of Substitution Mode LAST_VALUE.

7.3.2.4.2.3 Invocation of the Output Data object

For the invocation of the Output Data object the following rules apply:

- Output Data object elements (subslot) data shall not exceed the total length of 1 439 octets. The attributes List of Output Data Elements, Output Data Object Format respectively Format and Length have to be set accordingly.
- The relation of an Output Data object element to a Output CR, Input CR, M CR will be handled in the Context ASE.
- The sum of all Output Data object elements including IOPS and IOCS of the related Input Data object elements related to one CR shall not exceed the total length of 1 440 octets.
- One Output Data object shall be invoked in an application process of an IO device.

7.3.2.5 IO Data service specification**7.3.2.5.1 Local Set Input**

The Set Input service sets new input data of one slot/subslot together with its IOPS for next transmission for all related application relationships. The IOPS shall be at any time consistent with delivered Input Data values. The value of IOPS shall only be "GOOD" for valid input data of the whole subslot. In all other cases the value shall be set to "BAD_BY_xxx" and include the detecting instance e.g. subslot, slot or IO Device. This service is used by the application process of the IO device to update the addressed part of the protocol internal cyclic buffer. The actual transmission of the buffer is time triggered.

The Set Input service may set the IOPS only. It is the case if the slot/subslot does not contain Input Data.

NOTE 1 The local implementation of the service interface may support access variants with other than subslot granularity e.g. for more than one subslot or only parts (variables) of the Input Data object element. However, the Input Data object element is always consistent with its IOPS.

This service shall only be used in conjunction with the IO AR and Supervisor AR.

According to the cyclic buffer to buffer transportation characteristics the following behavior is possible:

- The Set Input requests are issued faster than the contents of the buffer is transported over the network. In this case only the values provided with the latest Set Input service are conveyed over the network.
- The Set Input requests are issued slower than the contents of the buffer is transported over the network. In this case the values provided with each Set Input service are conveyed more than once over the network.
- If the Set Input requests are issued synchronous with the transport of the contents of the buffers the values provided with each Set Input service are conveyed once over the network.

Table 116 shows the parameters of the service.

Table 116 – Local Set Input

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOPS	M	
Subslot Input Data	U	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

NOTE 2 The Set Input service sets the input data for all established ARs that belong to the submodule. In this case multiple calls for shared inputs are not necessary. Therefore, the AR and CR service parameter are omitted.

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

CREP

This parameter is the local identifier for the desired CR.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

IOPS

This parameter shall contain the values GOOD if data values are valid, BAD_BY_DEVICE, BAD_BY_SLOT, or BAD_BY_SUBSLOT.

Subslot Input Data

This parameter contains the value of the Input Data object element.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.2 Local Set Input IOCS

This service shall be used by a consumer (IO controller) of inputs to pass the value of changed IOCS to the application layer. The IOCS is conveyed back to the provider and provides a means to monitor the application. This service shall only be used in conjunction with the IO AR and Supervisor AR.

By this service the application process of the IO device updates the local buffer. According to the cyclic buffer to buffer transportation characteristics of the used application relationship the following behavior is possible:

- The Set Input IOCS requests are issued faster than the contents of the buffers are transported over the network to the Producer. In this case not each value of the IOCS is conveyed over the network. Only the latest value is transported, others are locally overwritten.
- The Set Input IOCS requests are issued slower than the contents of the buffers are transported over the network to the Producer. In this case each value of the IOCS provided with the service is conveyed more than once over the network.
- If the Set Input IOCS requests are issued synchronous with the transport of the contents of the buffers each value of the IOCS provided with the service is conveyed over the network.
- If there are several IOCS than only the IOCS of the non-shared CR will be indicated.

Table 117 shows the parameters of the service.

Table 117 – Local Set Input IOCS

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOCS	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

CREP

This parameter is the local identifier for the desired CR.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

IOCS

This parameter shall contain the values GOOD or BAD_BY_CONTROLLER.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.3 Local Get Input

This service shall only be used in conjunction with the IO AR and Supervisor AR. It is used by an IO controller to read the values of an input data subslot with the related IOPS. Table 118 shows the parameters of the service.

Table 118 – Local Get Input

Parameter name	req	cnt
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOPS		M
Subslot Input Data		C
New Flag		M
IOCS		C
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

CREP

This parameter is the local identifier for the desired CR.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

Result(+)

This parameter indicates that the service request succeeded.

IOPS

This parameter contains the value of the corresponding attribute of the Input Data object.

Subslot Input Data

This parameter contains the value of the Input Data object.

New Flag

This parameter contains the value TRUE if new Input data with IOPS have been received since last time calling the service. If the service was called more frequently the New Flag is set to FALSE by the protocol machines.

IOCS

This parameter contains the value of the IOCS in case there is corresponding Output Data object.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.4 Local Get Input IOCS

This service shall only be used in conjunction with the IO AR and Supervisor AR. It is used by a provider to get the IOCS of the IO controller. Table 119 shows the parameters of the service.

Table 119 – Local Get Input IOCS

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOCS		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

CREP

This parameter is the local identifier for the desired CR.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

Result(+)

This parameter indicates that the service request succeeded.

IOCS

This parameter contains the value of the IOCS.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.5 Local New Input

The New Input service signals to the application of the IO Controller a update of the input buffer after reception of a valid APDU. Table 120 shows the parameters of the service.

Table 120 – Local New Input

Parameter name	ind
Argument	M
AREP	M
CREP	M
Slot Number	M
Subslot Number	M
InData Flag	C

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR passed with the indication.

CREP

This parameter is the local identifier for the desired CR passed with the indication.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

InData Flag

This parameter indicates that the received buffer is usable or not.

NOTE All consumer IOCRs of the AR have got data first time.

7.3.2.5.6 Local Set Redundancy

This service shall be used by the IO device to set the “Redundancy Flag” of the selected AR. Table 121 shows the parameters of the service.

Table 121 – Local Set Redundancy

Parameter name	req	cnf
Argument	M	
AREP	U	
Redundancy Flag	U	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR passed with the request. It is omitted if the service parameter Provider State Flag is used because this parameter influences all established ARs transmitting IO Data.

Redundancy Flag

This parameter indicates with the value “primary present” that one AR within the AR set is in state “primary”. The value “no primary present” indicates that none of the AR within the AR set is seen as “primary”.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.7 Local Set State

This service shall be used by the IO device to set the “Redundancy Flag” of the selected AR. Table 122 shows the parameters of the service.

Table 122 – Local Set State

Parameter name	req	cnf
Argument	M	
AREP	U	
State Flag	U	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR passed with the request. It is omitted if the service parameter Provider State Flag is used because this parameter influences all established ARs transmitting IO Data.

State Flag

This parameter indicates if the AR is operating with the value „primary“ or standby with the value „backup“. The default value is „primary“.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.8 Local Data State Changed

The service signals a changed of the APDU Data Status value to the application of the IO controller or IO device. This service shall only be conveyed if the APDU Data Status value has been changed by the IO device or the IO controller and shall provide only the changed flags. Table 123 shows the parameters of the service.

Table 123 – Local Data State Changed

Parameter name	ind
Argument	M
AREP	M
CREP	M
Data Valid Flag	C
State Flag	C
Redundancy Primary Present Flag	C
Provider State Flag	C
Problem Indicator Flag	C

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR passed with the indication.

CREP

This parameter is the local identifier for the desired CR passed with the indication.

Data Valid Flag

This parameter indicates with the value "DataItem invalid" that all data (inclusive IOPS) of the AR are not valid and shall not be processed by the consumer.

The Data Flag with "DataItem invalid" could be used if the application process of the IO device is not able to set Input Data and/or IOPS due to a fatal application error. However, it should be used only in such serious cases because the Data Hold Time times out at the consumer and terminates the connection. In all other cases the IOPS should be used to tell the consumer about the status of the input data.

The value "DataItem valid" indicate that the data values including the IOPS, IOCS are valid.

State Flag

This parameter indicates if the AR is operating with the value "primary" or standby with the value "backup".

Redundancy Primary Present Flag

This parameter indicates with the value "primary present" that one AR within the AR set is "primary". The value "no primary present" indicates that none of the AR within the AR set is seen as "primary".

Provider State Flag

This parameter indicates if the application process is running with the value „run“ or stopped with the value „stop“. This value is only informative for the consumer application process, the actual validity of the data shall be according to IOPS.

NOTE 1 This flag could be used to control a global indicator to visualize the global remote application status.

Problem Indicator Flag

This parameter indicates if at least one diagnoses value is set within the Diagnoses ASE, which affects the related AR. It is indicated by the value „Problem detected“. It shall be set to „Regular operation“ as long as no diagnosis entry exists.

NOTE 2 This flag could be used to trigger the reading of diagnosis data within the IO controller. This flag does not say anything about the validity of input data that is only indicated by IOPS. A diagnosis entry may cause also to change the IOPS to “BAD” for local reasons by means of the Set Input service.

7.3.2.5.9 Read Record Input Data Object Element

This confirmed service may be used to read the value of an Input Data object element. This service can be used in conjunction with the IO AR, Supervisor AR, or implicit AR.

NOTE 1 A client application may use the service Read AR Data to read possible AR UUIDs of established ARs that could be used in this context.

Implicit AR using Target AR UUID: This service does only contain an Input Data object if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is responded.

Implicit AR using NIL AR UUID: This service does only contain an Input Data object if the application supports reading of values of Input Data objects without an established AR. Otherwise a parameter error is responded.

The service is optional for submodules which contain no input data.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 124 shows the specific elements of Data for this service.

Table 124 – Data elements of Read Record Input Data Object Element

Parameter name	service specific elements	rsp	cnf
Data	Length IOCS	M	M(=)
	IOCS	M	M(=)
	Length IOPS	M	M(=)
	IOPS	M	M(=)
	Length Data	M	M(=)
	Data	M	M(=)

Data

This parameter contains the values of the object which has been read.

Length IOCS

This parameter contains the value of the corresponding attribute of the Input Data object.

IOCS

This parameter contains the value of the corresponding attribute of the Input Data object.

Length IOPS

This parameter contains the value of the corresponding attribute of the Input Data object.

IOPS

This parameter contains the value of the corresponding attribute of the Input Data object.

Length Data

This parameter contains the data length of the object which has been read.

Data

This parameter contains the value of the object which has been read and consists of the number of octets indicated in the parameter Length Data. Its content is composed according to the attribute Input Data Structure of the Input Data object. Data contains the input data as given by the IO device.

7.3.2.5.10 Local Set Output

The Set Output service sets new output data of one slot/subslot together with its IOPS for next transmission for all related application relationships. The IOPS shall be at any time consistent with delivered Output Data values. The value of IOPS shall only be “GOOD” for valid output data of the whole subslot. In all other cases the value shall be set to “BAD_BY_CONTROLLER”. This service is used by the application process of the IO controller to update the addressed part of the protocol internal cyclic buffer. The actual transmission of the buffer is time triggered.

NOTE The local implementation of the service interface may support access variants with other than subslot granularity e.g. for more than one subslot or only parts (variables) of the Output Data object element. However, the Output Data object element is always consistent with its IOPS.

This service shall only be used in conjunction with the IO AR or Supervisor AR.

According to the cyclic buffer to buffer transportation characteristics the following behavior is possible:

- The Set Output requests issued faster than the contents of the buffers are transported over the network. In this case only the values provided with the latest Set Output service are conveyed over the network.
- The Set Output requests issued slower than the contents of the buffers are transported over the network. In this case the values provided with each Set Output service are conveyed more than once over the network.
- If the Set Output requests are issued synchronous with the transport of the contents of the buffers the values provided with each Set Output service are conveyed once over the network.

Table 125 shows the parameters of the service.

Table 125 – Local Set Output

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOPS	M	
Subslot Output Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

CREP

This parameter is the local identifier for the desired CR.

Slot Number

This parameter contains the slot number.

Subslot Number

This parameter contains the subslot number.

IOPS

This parameter shall contain the values GOOD if data values are valid or BAD_BY_CONTROLLER if this data are not valid.

Subslot Output Data

This parameter contains the value of the Output Data object element.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.11 Local Set Output IOCS

This service shall be used by a consumer (IO device) of outputs to pass the value of changed IOCS to the application layer. The IOCS is conveyed back to the provider and provides a means to monitor the application. This service shall only be used in conjunction with the IO AR and Supervisor AR.

By this service the application process of the IO device updates the local buffer. According to the cyclic buffer to buffer transportation characteristics of the used application relationship the following behavior is possible:

- The IO Status Change requests issued faster than the contents of the buffers are transported over the network to the Producer. In this case not each value of the IOCS is conveyed over the network. Only the latest value is transported, others are locally overwritten.
- The IO Status Change requests are issued slower than the contents of the buffers are transported over the network to the Producer. In this case each value of the IOCS provided with the service is conveyed more than once over the network.
- If the IO Status Change requests are issued synchronous with the transport of the contents of the buffers each value of the IOCS provided with the service is conveyed over the network.

Table 126 shows the parameters of the service.

Table 126 – Local Set Output IOCS

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOCS	M	

Parameter name	req	cnf
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

CREP

This parameter is the local identifier for the desired CR.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

IOCS

This parameter may contain the values GOOD, BAD_BY_DEVICE, BAD_BY_SLOT, or BAD_BY_SUBSLOT.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.12 Local Get Output

This service shall be used by an IO device to read the value for one slot/subslot of an Output Data object element together with its IOPS from the application layer. This service shall only be used in conjunction with the IO AR or Supervisor AR.

The IO device shall only process the values of the Output Data object element if the IOPS contains the value GOOD. Table 127 shows the parameters of the service.

Table 127 – Local Get Output

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOPS		M
Subslot Output Data		M

Parameter name	req	cnf
New Flag		M
IOCS		C
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR passed with the indication.

CREP

This parameter is the local identifier for the desired CR passed with the indication.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

Result(+)

This parameter indicates that the service request succeeded.

IOPS

This parameter contains the value of the corresponding attribute of the Output Data object element.

Subslot Output Data

This parameter contains the value of the Output Data object.

New Flag

This parameter contains the value TRUE if new Output data with IOPS have been received since last time calling the service. If the service was called more frequently the New Flag is set to FALSE by the protocol machines.

IOCS

This parameter contains the value of the IOCS in case there is a corresponding Input Data object.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.13 Local Get Output IOCS

This service shall be used by an IO device to read the IOCS from the application layer. This service shall only be used in conjunction with the IO AR or Supervisor AR. Table 128 shows the parameters of the service.

Table 128 – Local Get Output IOCS

Parameter name	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	

Parameter name	req	cnf
Result(+)		S
AREP		M
IOCS		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR passed with the indication.

CREP

This parameter is the local identifier for the desired CR passed with the indication.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

Result(+)

This parameter indicates that the service request succeeded.

IOCS

This parameter contains the value of the IOCS.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.14 Local New Output

The New Output service signals to the application of the IO device a update of the output buffer after reception of a valid APDU. Table 129 shows the parameters of the service.

Table 129 – Local New Output

Parameter name	ind
Argument	M
AREP	M
CREP	M
Slot Number	M
Subslot Number	M
InData Flag	M

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR passed with the indication.

CREP

This parameter is the local identifier for the desired CR passed with the indication.

Slot Number

This parameter contains the local slot number.

Subslot Number

This parameter contains the local subslot number.

InData Flag

This parameter indicates that the received buffer is usable or not.

NOTE All consumer IOCRs of the AR have got data first time.

7.3.2.5.15 Local Set Provider State

This service shall be used by the IO controller to set the “Provider State Flag” in the field APDU Data Status of the next conveyed APDU. The flag “Provider State Flag” influences all established ARs of the IO device. Table 130 shows the parameters of the service.

Table 130 – Local Set Provider State

Parameter name	req	cnf
Argument	M	
AREP	U	
Provider State Flag	U	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR passed with the request. It is omitted if the service parameter Provider State Flag is used because this parameter influences all established ARs transmitting IO Data.

Provider State Flag

This parameter indicates if the application process is running with the value „run“ or stopped with the value „stop“. This value is only informative for the consumer application process, the actual validity of the data shall be according to IOPS. In case of „stop“ all IOPS shall also be set to the value “BAD_BY_CONTROLLER” by means of the Set Input service request to all slots/subslots.

NOTE This flag could be used to control a global indicator to visualize the global remote application status.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.2.5.16 Read Record Output Data Object Element

This confirmed service may be used to read the value of an Output Data object element with the substitute value. This service can be used in conjunction with the IO AR, Supervisor AR, or the implicit AR.

NOTE 1 A client application may use the service Read AR Data to read possible AR UUIDs of established ARs that could be used in this context.

Implicit AR using Target AR UUID: This service does only contain an Output Data object if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is responded.

Implicit AR using NIL AR UUID: This service does only contain an Output Data object if the application supports reading of values of Output Data objects without an established AR. Otherwise a parameter error is responded.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 131 shows the specific elements of Data for this service.

Table 131 – Data elements of Read Record Output Data Object Element

Parameter name	service specific elements	rsp	cnf
Data	Substitute Active Flag	M	M(=)
	Length IOCS	M	M(=)
	Length IOPS	M	M(=)
	Length Data	M	M(=)
	Data Item	M	M(=)
	IOCS	M	M(=)
	Data	M	M(=)
	IOPS	M	M(=)
	Substitute Value	M	M(=)
	Substitution Mode	M	M(=)
	Substitute Data Item	M	M(=)
	IOCS	M	M(=)
	Data	M	M(=)
	Substitute Data Valid	M	M(=)

Data

This parameter contains the values of the object which has been read.

Substitute Active Flag

This parameter determines which data is applied to the output.

If this parameter is set to FALSE, the submodule carries the values according to the Data Item.

If this parameter is set to TRUE, the submodule carries

- the values according to the Substitute Data Item if Substitution Mode set to LAST_VALUE, REPLACEMENT_VALUE or PROFILE_SPECIFIC,
- data set to “all zero” if Substitution Mode is set to ZERO.

Length IOCS

This parameter contains the value of the corresponding attribute of the Output Data object.

Length IOPS

This parameter contains the value of the corresponding attribute of the Output Data object.

Length Data

This parameter contains the data length of the object which has been read.

Data Item

This parameter is composed of the following elements:

IOCS

This parameter contains the value of the corresponding attribute of the Output Data object.

Data

This parameter contains the value of the object which has been read and consists of the number of octets indicated in the Length Data of the response. Its content is composed according to the attribute Output Data Structure of the Output Data object. Data contains the output data as given by the IO controller.

IOPS

This parameter contains the value of the corresponding attribute of the Output Data object.

Substitute Value

This parameter is composed of the following elements:

Substitution Mode

This parameter contains the value of the corresponding attribute of the Output Data object.

Substitute Data Item

This parameter is composed of the following elements:

IOCS

This parameter contains the value of the corresponding attribute of the Output Data object.

Data

This parameter contains the value of the object which has been read and consists of the number of octets indicated in the Length Data of the response. Its content is composed according to the attribute Output Data Object Format of the Output Data object. Data contains the substitute data as given by the IO controller.

Substitute Data Valid

This parameter contains the value of the corresponding attribute of the Output Data object.

7.3.2.5.17 Read Substitute Value

This confirmed service may be used to read the substitute value. This service can be used in conjunction with the IO AR, Supervisor AR, or the implicit AR.

NOTE 1 A client application may use the service Read AR Data to read possible AR UUIDs of established ARs that could be used in this context.

Implicit AR using Target AR UUID: This service does only contain a Substitute Value object if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is responded.

Implicit AR using API: This service does only contain a Substitute Value object if the application supports reading of values of Substitute Value objects without an established AR. Otherwise a parameter error is responded.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 132 shows the specific elements of Data for this service.

Table 132 – Data elements of Read Substitute Value

Parameter name	service specific elements	rsp	cnf
Data	Substitute Value	M	M(=)
	Substitution Mode	M	M(=)
	Substitute Data Item	M	M(=)
	IOCS	M	M(=)
	Data	M	M(=)
	Substitute Data Valid	M	M(=)

Data

This parameter contains the values of the object which has been read.

Substitute Value

This parameter is composed of the following elements:

Substitution Mode

This parameter contains the value of the corresponding attribute of the Output Data object.

Substitute Data Item

This parameter is composed of the following elements:

IOCS

This parameter contains the value of the corresponding attribute of the Output Data object.

Data

This parameter contains the Substitute Value which has been read and consists of the number of octets indicated in the Length Data of the response. Its content is composed according to the attribute Output Data Object Format of the Output Data object.

Substitute Data Valid

This parameter contains the value of the corresponding attribute of the Output Data object.

If the submodule does not support Substitute Value either a negative response containing the values Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" may be given or else a null record may be returned.

7.3.2.5.18 Write Substitute Value

This confirmed service may be used to write the substitute value of an Output Data object element. This service can be used in conjunction with the IO AR and Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 133 shows the specific elements of Data for this service.

Table 133 – Data elements of Write Substitute Value

Parameter name	service specific elements	req	ind
Data	Substitute Value	M	M(=)
	Substitution Mode	M	M(=)
	Substitute Data Item	M	M(=)
	IOCS	M	M(=)
	Data	M	M(=)
	Substitute Data Valid	M	M(=)

Data

This parameter contains the value of the object which may be used to write.

Substitute Value

This parameter is composed of the following elements:

Substitution Mode

This parameter contains the value of the corresponding attribute of the Output Data object.

Substitute Data Item

This parameter is composed of the following elements:

IOCS

This parameter contains the value of the corresponding attribute of the Output Data object.

Data

This parameter contains the Substitute Data and consists of the number of octets indicated in the Length Data of the response. Its content is composed according to the attribute Output Data Object Format of the Output Data object.

Substitute Data Valid

This parameter contains the value of the corresponding attribute of the Output Data object.

7.3.2.6 Behavior of IO Data objects**7.3.2.6.1 General behavior of the Input Data object**

For performance reasons, the value of the Input Data object shall be transferred on change as fast as possible to the IO AR buffer for transmission. The speed and the jitters of this action is a performance parameter of an IO device.

7.3.2.6.2 General behavior of the Output Data object

For performance reasons, the value for the Output Data object shall be transferred on receipt as fast as possible to the object. The speed and the jitters of this action is a performance parameter of a IO device. This parameter determines the synchronization accuracy.

The Read Output service shall provide the current value of the application process which has been fetched with the Get Output service.

NOTE Due to different reasons (e.g. physical output delays) this value may differ from the value of the real output data or from the last transferred value.

7.3.3 LogBook Data ASE

7.3.3.1 Overview

The LogBook Data ASE provides a mean to store locally generated and AR specific events in a circular buffer and provides a service to read that event buffer. The content of entries is manufacturer specific and may be used to identify communication problems e.g. reasons for AR termination.

The LogBook Data ASE defines attributes of LogBook Data objects and provides a set of services used to read remotely, and write their values locally. A LogBook Data ASE is specific for an IO Device. The LogBook Data object is addressed implicitly by the defined services.

The access to LogBook Data objects is performed according to the Client/Server access model for read. The Client/Server model is characterized by a Client application sending a read request to a Server application that responds accordingly.

The formal model of the LogBook Data ASE is presented by the LogBook Data class specification, containing a description of its attributes, services as well as invocations, followed by a detailed service specification.

7.3.3.2 LogBook Data class specification

7.3.3.2.1 Template

A LogBook Data object is described by the following template:

ASE:	LogBook Data ASE
CLASS:	LogBook Data
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: Current Local Time Stamp
3	(m) Attribute: List of Entries
3.1	(m) Attribute: Local Time Stamp
3.2	(m) Attribute: AR UUID
3.3	(m) Attribute: PNIO Status
3.4	(m) Attribute: Entry Detail
SERVICES:	
1	(m) OpsService: Read LogBook Data
2	(m) OpsService: Local Create LogBook Entry

7.3.3.2.2 Attributes

Implicit

The key attribute Implicit indicates that the LogBook Data object is implicitly addressed by the services.

Current Local Time Stamp

This attribute contains the current local time by means of the value of the cycle counter.

Attribute type: Unsigned64

List of Entries

An IO Device shall support at least 16 LogBook entries and an IO Controller shall support at least 4 KByte. This attribute list consists of the following attributes.

Local Time Stamp

This attribute contains the local time when the entry was made by means of the value of the cycle counter at this point of time.

Attribute type: Unsigned64

AR UUID

This attribute defines the UUID of the IO AR entry which is the source.

Attribute type: UUID

PNIO Status

This attribute contains the protocol machine defined value of the entry.

Attribute type: Unsigned32

Allowed Values: see IEC 61158-6-10

Entry Detail

This attribute contains the protocol machine defined value of the entry.

Attribute type: Unsigned32

7.3.3.3 LogBook Data service specification**7.3.3.3.1 Read LogBook Data**

This confirmed service may be used to read the value of a LogBook Data object. This service can be used in conjunction with the implicit AR, IO AR or Supervisor AR.

If the actual log book contains more bytes than the requested length, only the newest entries (according to the requested amount of entries) shall be used for the response. The returned list does not need to be sorted timely; the requestor of the information is able to do this with the help of the local time stamps.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4 1.1). But the parameter Data has a service specific structure and meaning. Table 134 shows the specific elements of Data for this service.

Table 134 – Data elements of Read LogBook Data

Parameter name	service specific elements	rsp	cnf
Data	Current Local Time Stamp	M	M(=)
	List of Entries	M	M(=)
	Local Time Stamp	M	M(=)
	AR UUID	M	M(=)
	PNIO Status	M	M(=)
	Entry Detail	M	M(=)

Data

This parameter contains the values of the object which has been read.

Current Local Time Stamp

This parameter contains the value of the corresponding attribute of the ASE object.

List of Entries

This parameter is composed of the following list elements:

Local Time Stamp

This parameter contains the value of the corresponding attribute of the ASE object.

AR UUID

This parameter contains the value of the corresponding attribute of the ASE object.

PNIO Status

This parameter contains the value of the corresponding attribute of the ASE object.

Entry Detail

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.3.3.2 Local Create LogBook Entry

This local service indicates value for the LogBook Data object. Table 135 shows the parameters of the service.

Table 135 – Local Create LogBook Entry

Parameter name	ind
Argument	M
AREP	M
Local Time Stamp	M
AR UUID	M
PNIO Status	M
Entry Detail	M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Local Time Stamp

This parameter contains the value of the corresponding attribute of the ASE object.

AR UUID

This parameter contains the value of the corresponding attribute of the ASE object.

PNIO Status

This parameter contains the value of the corresponding attribute of the ASE object.

Entry Detail

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.4 Diagnosis ASE

7.3.4.1 Overview

In the application layer environment, the Diagnosis ASE of an IO device contains different diagnosis entries that are used to keep the diagnosis information of the IO device. The diagnosis information from the IO device is related to the addressed AP and includes only those slots, subslots and channels which have set diagnosis.

NOTE If a subslot or channel does not have diagnosis set then this information is not transmitted.

The Diagnosis ASE defines attributes of the Diagnosis objects and provides a set of services used to read their values. The following diagnosis types are defined:

- Channel Diagnosis
- Extended Channel Diagnosis
- Qualified Channel Diagnosis
- Manufacturer Specific Diagnosis

It is strongly recommended to use Channel Diagnosis, Ext Channel Diagnosis, or Qualified Channel Diagnosis in all matching cases because detailed semantics are fully standardized. Only if the above not fits, manufacturer specific diagnosis data can be conveyed by means of the Manufacturer Specific Diagnosis.

Note that channel diagnosis, extended channel diagnosis and qualified channel diagnosis differ just in the amount of data provided by the IO device for the specific diagnosis item. Channel diagnosis offers channel properties and channel error type. Extended channel diagnosis adds an extended channel error type and an additional value. Qualified channel diagnosis adds the qualified channel qualifier. What format is chosen for a particular diagnosis entry is decided by the manufacturer of the IO device. Thus always the same format could be chosen (e.g. using always the extended channel diagnosis format).

Manufacturer specific diagnosis offers channel properties, a user structure identifier (to specify the content of the data following) and plain data.

The severity is defined as diagnosis, maintenance demanded, maintenance required and status. The classification of diagnosis, maintenance and qualified according to their severity is shown in Figure 27.

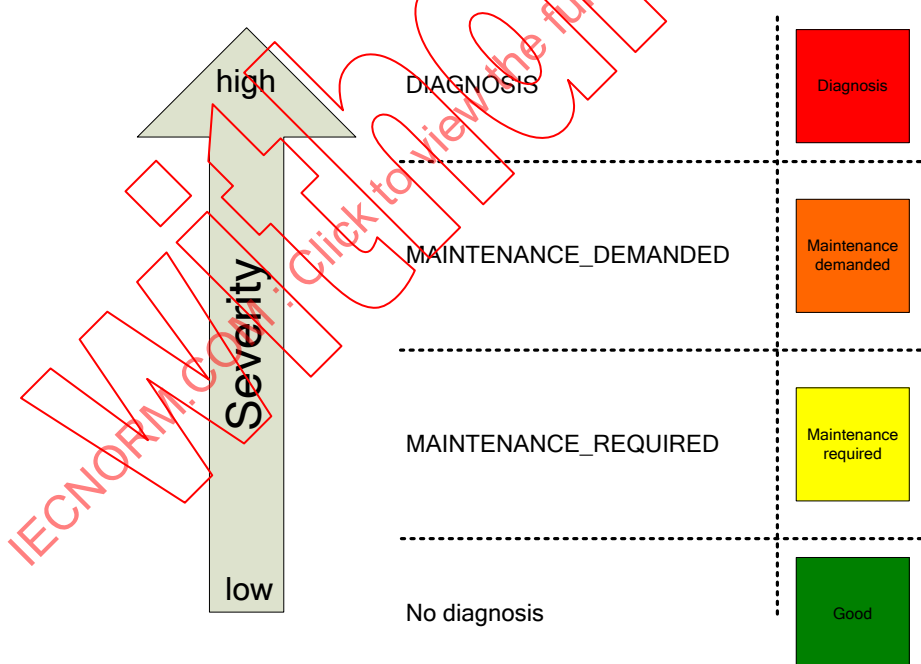


Figure 27 – Severity classification of diagnosis and maintenance

The severity of a channel diagnosis is expressed via the attribute ChannelProperties.Maintenance.

If the attribute is set to DIAGNOSIS, MAINTENANCE_DEMANDED, or MAINTENANCE_REQUIRED the severity is set according to the value of the attribute (see Figure 27).

NOTE The IO device and the IO controller may derive user signalling (e.g. by LEDs) from this severity (as indicated in Figure 27).

If the attribute is set to the value QUALIFIED_DIAGNOSIS, the meaning and the severity is derived from the QualifiedChannelQualifier by application means. The qualifiers 2 up to 31 within that attribute are classified by a profile. Thus the value QUALIFIED_DIAGNOSIS is not seen as severity by both IO device and IO controller. A higher level of diagnosis (e.g. maintenance station) may derive a severity from the qualifiers.

The severity “status” is provided solely for manufacturer specific diagnosis. Manufacturer Specific Diagnosis indicating diagnosis, maintenance demanded and maintenance required have ChannelProperties.Specifier set to APPEARS. In contrary Manufacturer Specific Diagnosis indicating a status have ChannelProperties.Maintenance set to DIAGNOSIS and ChannelProperties.Specifier set to DISAPPEARS.

The application of an IO device may provide the value of one or several Diagnosis objects to the implicit AR, IO AR, and the Supervisor AR.

The Diagnosis model uses the Client/Server access model.

Furthermore, in case of setting the first diagnosis entry the Input Data ASE shall be informed by means of the event (Diagnosis ASE "First Diagnosis Entry set"). In case of unsetting the last diagnosis entry the Input Data ASE shall be informed by means of the event (Diagnosis ASE "Last Diagnosis Entry cleared"). The Input Data ASE shall not be informed if only maintenance information is set or cleared.

The Alarm ASE may also be informed to send alarms of those alarm types defined in Table 161 and marked with “Attached to Diagnosis ASE = Yes” if it is configured by the application.

7.3.4.2 Diagnosis class specification

7.3.4.2.1 Template

The Diagnosis object is described by the following template:

ASE:	Diagnosis ASE
CLASS:	Diagnosis
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: Device Status
3	(c) Constraint: Device Status <> DEVICE_OK
3.1	(m) Attribute: List of APs
3.1.1	(m) Attribute: API
3.1.2	(m) Attribute: List of Slots
3.1.2.1	(m) Attribute: Slot Number
3.1.2.2	(m) Attribute: List of Subslots
3.1.2.2.1	(m) Attribute: Subslot Number
3.1.2.2.2	(m) Attribute: Submodule Diagnosis State
3.1.2.2.3	(m) Attribute: List of Channels
3.1.2.2.3.1	(m) Attribute: Channel Number
3.1.2.2.3.2	(m) Attribute: List of Channel Diagnosis
3.1.2.2.3.2.1	(m) Attribute: Time Stamp
3.1.2.2.3.2.2	(m) Attribute: Channel Properties
3.1.2.2.3.2.2.1	(m) Attribute: Type

3.1.2.2.3.2.2.2	(m) Attribute:	Accumulative
3.1.2.2.3.2.2.3	(m) Attribute:	Maintenance
3.1.2.2.3.2.2.4	(m) Attribute:	Specifier
3.1.2.2.3.2.2.5	(m) Attribute:	Direction
3.1.2.2.3.2.3	(m) Attribute:	User Structure Identifier
3.1.2.2.3.2.4	(o) Attribute:	Channel Error type
3.1.2.2.3.2.5	(o) Attribute:	Ext Channel Error type
3.1.2.2.3.2.6	(o) Attribute:	Ext Channel Add Value
3.1.2.2.3.2.7	(o) Attribute:	Qualified Channel Qualifier
3.1.2.2.3.2.8	(o) Attribute:	List of Manufacturer Specific Diagnosis
3.1.2.2.3.2.8.1	(m) Attribute:	Data
3.2	(o) Attribute:	List of Nested Diagnosis Info
3.2.1	(m) Attribute:	IP Address
3.2.2	(m) Attribute:	Station Name Length
3.2.3	(m) Attribute:	Name Of Station Value

SERVICES:

1	(m) OpsService:	Local Add Diagnosis Entry
2	(m) OpsService:	Local Remove Diagnosis Entry
3	(m) OpsService:	Local Diagnosis Event
4	(m) OpsService:	Read Diagnosis in channel coding for one subslot
5	(m) OpsService:	Read Diagnosis in all codings for one subslot
6	(m) OpsService:	Read Maintenance required in channel coding for one subslot
7	(m) OpsService:	Read Maintenance required in all codings for one subslot
8	(m) OpsService:	Read Maintenance demanded in channel coding for one subslot
9	(m) OpsService:	Read Maintenance demanded in all codings for one subslot
10 subslot	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one
11	(m) OpsService:	Read Diagnosis in channel coding for one slot
12	(m) OpsService:	Read Diagnosis in all codings for one slot
13	(m) OpsService:	Read Maintenance required in channel coding for one slot
14	(m) OpsService:	Read Maintenance required in all codings for one slot
15	(m) OpsService:	Read Maintenance demanded in channel coding for one slot
16	(m) OpsService:	Read Maintenance demanded in all codings for one slot
17	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one slot
18	(m) OpsService:	Read Diagnosis in channel coding for one AR
19	(m) OpsService:	Read Diagnosis in all codings for one AR
20	(m) OpsService:	Read Maintenance required in channel coding for one AR
21	(m) OpsService:	Read Maintenance required in all codings for one AR
22	(m) OpsService:	Read Maintenance demanded in channel coding for one AR
23	(m) OpsService:	Read Maintenance demanded in all codings for one AR
24	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one AR
25	(m) OpsService:	Read Diagnosis in channel coding for one API
26	(m) OpsService:	Read Diagnosis in all codings for one API
27	(m) OpsService:	Read Maintenance required in channel coding for one API
28	(m) OpsService:	Read Maintenance required in all codings for one API
29	(m) OpsService:	Read Maintenance demanded in channel coding for one API
30	(m) OpsService:	Read Maintenance demanded in all codings for one API
31	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one API
33 device	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one
34	(o) OpsService:	Read Nested Diagnosis Info

7.3.4.2.2 Attributes

Implicit

The key attribute Implicit indicates that the Diagnosis object is implicitly addressed by the services.

Device Status

This attribute contains the diagnosis summary. If the value is set to DEVICE_OK no slot/subslot shall contain diagnosis information. If the value is set to DEVICE_FAILURE, MAINTENANCE, or MANUFACTURER_SPECIFIC_STATUS at least one slot/subslot shall contain diagnosis information, maintenance or manufacturer specific status information.

Attribute type: Unsigned8

Allowed values: DEVICE_OK, DEVICE_FAILURE, MAINTENANCE or MANUFACTURER_SPECIFIC_STATUS

List of APs

This attribute contains the diagnosis, maintenance, qualified or manufacturer specific status information specific for each supported application process. A list element consists of the following attributes:

API

This attribute shall consist only of APIs having diagnosis, maintenance, qualified, or manufacturer specific status information.

List of Slots

This attribute contains only slots having diagnosis, maintenance, qualified, or manufacturer specific status information. A list element consists of the following attributes.

Slot Number

This attribute defines the slot number as a local identifier. This number is unique within a device. The numbering of slots within a device shall be in ascending order but may contain gaps in order to allow logical structuring. The list shall contain all physical or logical slots having diagnosis, maintenance, qualified, or manufacturer specific status information.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

List of Subslots

This attribute contains only subslots having diagnosis, maintenance, qualified, or manufacturer specific status information. A list element consists of the following attributes.

Subslot Number

This attribute defines the subslot number as a local identifier. This number is unique within a slot. The numbering of subslots within a slot shall be in ascending order but may contain gaps in order to allow logical structuring. The list shall contain all physical or logical subslots having diagnosis, maintenance, qualified, or manufacturer specific status information.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF are allowed

NOTE 1 Subslot Number 0 is used in conjunction with AlarmType "Pull" to address a certain module within the IO device; it never contains a diagnosis.

Submodule Diagnosis State

This attribute contains the diagnosis summary for the submodule. The value SUBMODULE_FAILURE shall only be set in case of at least on Channel Diagnosis severity other than INFORMATION and at least one Manufacturer Specific Diagnosis with severity other than INFORMATION and channel properties specifier

shall be set to APPEARS. In all other cases the value shall be set to SUBMODULE_OK.

Attribute type: Unsigned8

Allowed values: SUBMODULE_OK, SUBMODULE_FAILURE

List of Channels

This attribute contains only channels having diagnosis, maintenance, qualified, or manufacturer specific status information. A list element consists of the following attributes.

Channel Number

This attribute defines the channel number as a local identifier. This number is unique within a subslot. The numbering of channels within a subslot shall be in ascending order but may contain gaps in order to allow logical structuring. The list shall contain all physical or logical channels having diagnosis, maintenance, qualified, or manufacturer specific status information.

The special value 0x8000 identifies the whole submodule instead of a special channel.

In these cases, the attribute Channel Properties Accumulative shall contain the value INDIVIDUAL.

If the attribute Channel Properties Accumulative contains the value CHANNELGROUP then this attribute contains the number of the lowest affected channel.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF user specific channel number, 0x8000 submodule

List of Channel Diagnosis

This list contains all diagnosis, maintenance, qualified, manufacturer specific diagnosis, or manufacturer specific status information related to this channel. A list element consists of the following attributes:

Time Stamp

This attribute contains the time when this diagnosis, maintenance, qualified, manufacturer specific diagnosis, or manufacturer specific status information was detected.

Attribute type: TimeStamp

Channel Properties

This attribute consists of the following attributes:

Type

This attribute indicates the channel's data length to which the Channel Diagnosis object is related.

Allowed values:

1BIT, 2BIT, 4BIT, BYTE, WORD, 2WORDS, 4WORDS, UNSPECIFIC

UNSPECIFIC shall be used if the Channel Number contains the value 0x8000 or the data length is different than all other allowed values.

Accumulative

This attribute indicates if the diagnosis is related to a certain channel or to a group of channels.

Allowed values: INDIVIDUAL, CHANNELGROUP

Maintenance

This attribute indicates the severity of the diagnosis.

The value DIAGNOSIS is used to indicate diagnosis including manufacturer specific diagnosis and status.

The value MAINTENANCE_REQUIRED and MAINTENANCE_DEMANDED are used to indicate maintenance state.

NOTE 2 For example, a printer indicates “toner low” as a maintenance requirement (e.g. to order a new toner cartridge) and “toner low, print quality affected” as a maintenance demand (e.g. to replace the toner cartridge).

The value QUALIFIED_DIAGNOSIS is used to indicate qualified diagnosis.

Allowed values: DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS

Specifier

This attribute indicates the diagnosis specifier.

Allowed values:

APPEARS – for diagnosis, maintenance, qualified, or manufacturer specific diagnosis

DISAPPEARS – for manufacturer specific status information

NOTE 3 For example, a printer indicates “toner empty, printer stopped” as a diagnosis appears.

Direction

This attribute indicates the direction of the channel to which the Channel Diagnosis object is related.

Attribute type: Unsigned8

Allowed values: INPUT_CHANNEL, OUTPUT_CHANNEL, BIDIRECTIONAL_CHANNEL, MANUFACTURER_SPECIFIC

User Structure Identifier

This parameter indicates the format of the following data.

Attribute type: Unsigned16

Allowed values: CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, QUALIFIED_CHANNEL_DIAGNOSIS_USI, MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF

NOTE 4 The IO device determines in which format a specific diagnosis item is returned. The requestor is able to determine, whether manufacturer specific data shall be returned (by requesting “all codings”) or not (by requesting “channel coding”).

Channel Error type

This optional attribute indicates the error type of the Channel Related Diagnosis. It is present only, if the attribute User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Attribute type: Unsigned16

Table 136 – Channel Error Type

Channel Error Type	Meaning
SHORT_CIRCUIT	Short circuit
UNDERVOLTAGE	Undervoltage
OVERVOLTAGE	Overvoltage
OVERLOAD	Overload
OVERTEMPERATURE	Overtemperature
LINE_BREAK	Line break
UPPER_LIMIT_VALUE_EXCEEDED	Upper limit value exceeded
LOWER_LIMIT_VALUE_EXCEEDED	Lower limit value exceeded
ERROR	Error
SIMULATION_ACTIVE	Simulation active
PARAMETER_MISSING	Parameter missing
PARAMETERIZATION_FAULT	Parameterization fault
POWER_SUPPLY_FAULT	Power supply fault
FUSE_BLOWN_OR_OPEN	Fuse blown / open
COMMUNICATION_FAULT	Communication fault
GROUND_FAULT	Ground fault
REFERENCE_POINT_LOST	Reference point lost
PROCESS_EVENT_LOST	Process event lost / sampling error
THRESHOLD_WARNING	Threshold warning
OUTPUT_DISABLED	Output disabled
SAFETY_EVENT	Safety event
EXTERNAL_FAULT	External fault
TEMPORARY_FAULT	Temporary fault
DATA_TRANSMISSION_IMPOSSIBLE	Data transmission impossible
REMOTE_MISMATCH	Remote mismatch
MEDIA_REDUNDANCY_MISMATCH	Media redundancy mismatch
SYNC_MISMATCH	Sync mismatch
ISOCRONOUS_MODE_MISMATCH	IsochronousMode mismatch
MULTICAST_CR_MISMATCH	Multicast CR mismatch
FIBER_OPTIC_MISMATCH	Fiber optic mismatch
NETWORK_COMPONENT_FUNCTION_MISMATCH	Network component function mismatch
TIME_MISMATCH	Time mismatch
DYNAMIC_FRAME_PACKING_FUNCTION_MISMATCH	Dynamic frame packing function mismatch
MEDIA_REDUNDANCY_WITH_PLANNED_DUPLICATION_MISMATCH	Media redundancy with planned duplication mismatch
SYSTEM_REDUNDANCY_MISMATCH	System redundancy mismatch
MULTIPLE_INTERFACE_MISMATCH	Multiple interface mismatch
NESTED_DIAGNOSIS_INDICATION	Nested diagnosis indication
MANUFACTURER_SPECIFIC_001B – MANUFACTURER_SPECIFIC_001E, MANUFACTURER_SPECIFIC_0100 – MANUFACTURER_SPECIFIC_7FFF	Manufacturer specific

Channel Error Type	Meaning
PROFILE_SPECIFIC_0020 up to PROFILE_SPECIFIC_00FF	Common profiles
PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF	Profiles

Ext Channel Error type

This optional attribute depends on the attribute Channel Error type and indicates an extended error type, where applicable. It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI. The allowed values are shown in Table 137.

Attribute type: Unsigned16

Table 137 – Ext Channel Error type

Channel Error Type	Ext Channel Error Type
SHORT_CIRCUIT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
UNDERVOLTAGE	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
OVERVOLTAGE	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
OVERLOAD	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
OVERTEMPERATURE	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
LINE_BREAK	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
UPPER_LIMIT_VALUE_EXCEEDED	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO

Channel Error Type	Ext Channel Error Type
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
LOWER_LIMIT_VALUE_EXCEEDED	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
ERROR	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
PARAMETER_MISSING	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
SIMULATION_ACTIVE	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
PARAMETERIZATION_FAULT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
POWER_SUPPLY_FAULT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
FUSE_BLOWN_OR_OPEN	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
COMMUNICATION_FAULT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
GROUND_FAULT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF

Channel Error Type	Ext Channel Error Type
REFERENCE_POINT_LOST	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
PROCESS_EVENT_LOST	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
THRESHOLD_WARNING	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
OUTPUT_DISABLED	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
SAFETY_EVENT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
EXTERNAL_FAULT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
TEMPORARY_FAULT	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
DATA_TRANSMISSION_IMPOSSIBLE	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	LINK_STATE_MISMATCH_LINK_DOWN
	MAU_TYPE_MISMATCH
	LINE_DELAY_MISMATCH
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
REMOTE_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	PEER_CHASSIS_ID_MISMATCH
	PEER_PORT_ID_MISMATCH
	PEER_RT_CLASS_3_MISMATCH

Channel Error Type	Ext Channel Error Type
	PEER_MAUTYPE_MISMATCH
	PEER_MRP_DOMAIN_MISMATCH
	NO_PEER_DETECTED
	PEER_LINE_DELAY_MISMATCH
	PEER_PTCP_MISMATCH
	PEER_PREAMBLE_LENGTH_MISMATCH
	PEER_FRAGMENTATION_MISMATCH
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
MEDIA_REDUNDANCY_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	MANAGER_ROLE_FAIL
	MRP_RING_OPEN
	MULTIPLE_MANAGER
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
SYNC_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	NO_SYNC_MESSAGE_RECEIVED
	JITTER_OUT_OF_BOUNDARY
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
ISOCRONOUS_MODE_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	OUTPUT_TIME_FAILURE
	INPUT_TIME_FAILURE
	MASTER_LIFE_SIGN_FAILURE
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
MULTICAST_CR_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	MULTICAST_CONSUMER_CR_TIMED_OUT
	ADDRESS_RESOLUTION_FAILED
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
FIBER_OPTIC_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	POWER_BUDGET
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
NETWORK_COMPONENT_FUNCTION_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	FRAME_DROPPED_NO_RESOURCE
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF

Channel Error Type	Ext Channel Error Type
DYNAMIC_FRAME_PACKING_FUNCTION_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	FRAME_LATE_ERROR_FOR_FRAME_0100 up to FRAME_LATE_ERROR_FOR_FRAME_0FFF
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
MEDIA_REDUNDANCY_WITH_PLANNED_DUPLICATION_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	MRPD_DUPLICATION_VOID_FOR_FRAME_0700 up to MRPD_DUPLICATION_VOID_FOR_0 FRAME_0FFF
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
SYSTEM_REDUNDANCY_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	SYSTEM_REDUNDANCY_EVENT
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
MULTIPLE_INTERFACE_MISMATCH	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	STANDARD_GATEWAY_MISMATCH
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
NESTED_DIAGNOSIS_INDICATION	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	SUBORDINATED_ENTITY_INFORMATION
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF
MANUFACTURER_SPECIFIC_001B – MANUFACTURER_SPECIFIC_001E, MANUFACTURER_SPECIFIC_0100 – MANUFACTURER_SPECIFIC_7FFF	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_90FF	MANUFACTURER_SPECIFIC_0001 up to MANUFACTURER_SPECIFIC_7FFF
	PROFILE_SPECIFIC_9000 up to PROFILE_SPECIFIC_9FFF

The allowed combinations of Channel Error Type and Ext Channel Error Type are shown in Table 138.

Table 138 – Allowed combinations of Channel Error Type and Ext Channel Error Type

ChannelErrorType	ExtChannelErrorType	Usage
Normative	Normative	Allowed
Normative	Vendor specific	Allowed
Normative	Profile specific	Allowed
Vendor specific	Normative	Prohibited
Vendor specific	Vendor specific	Allowed
Vendor specific	Profile specific	Prohibited

ChannelErrorType	ExtChannelErrorType	Usage
Profile specific	Normative	Prohibited
Profile specific	Vendor specific	Allowed
Profile specific	Profile specific	Allowed

Ext Channel Add Value

This optional attribute contains an additional value to further specify the extended channel error. It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

In case the attribute Ext Channel Error type contains the value ACCUMULATIVE_INFO then the value of the attribute shall have the meaning according to Table 139.

Otherwise the value is defined by means of the ext channel-error type.

Table 139 – Ext Channel Add Value for Accumulative Info

Value	Meaning
Bit0: 0x00	ChannelNumber is not affected
Bit0: 0x01	ChannelNumber is affected
Bit1: 0x00	ChannelNumber +1 is not affected
Bit1: 0x01	ChannelNumber +1 is affected
Bit2: 0x00	ChannelNumber +2 is not affected
Bit2: 0x01	ChannelNumber +2 is affected
...	...
Bit30: 0x00	ChannelNumber +30 is not affected
Bit30: 0x01	ChannelNumber +30 is affected
Bit31: 0x00	ChannelNumber +31 is not affected
Bit31: 0x01	ChannelNumber +31 is affected

Attribute type: Unsigned32

Qualified Channel Qualifier

This optional attribute contains the qualified diagnosis data. It is present only, if the attribute User Structure Identifier is set to QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Bit 0 and Bit 1 are reserved. Bit 2 indicates whether Qualifier_2 is set, Bit 3 whether Qualifier_3 is set, ... and Bit 31 whether Qualifier_31 is set. The values shall be used profile specific.

Attribute type: Unsigned32

List of Manufacturer Specific Diagnosis Data

This optional attribute contains the manufacturer specific diagnosis data, or manufacturer specific status information.

This attribute is present only, if the attribute User Structure Identifier contains a value in the range from MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF

A list element consists of the following attributes:

Data

This attribute contains manufacturer specific diagnosis or status data.

Attribute type: One of the following data types shall be used:
Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference,
Floating Point, Network Time, VisibleString, Network Time Difference,
OctetString

List of Nested Diagnosis Info

This optional attribute contains the list of subordinated devices having diagnosis, maintenance, qualified, manufacturer specific diagnosis, or manufacturer specific status information. The optional attribute exists only in case the device indicates channel error type NESTED_DIAGNOSIS_INDICATION.

Each list element indicates an IO device in the subordinated IO system of this device having diagnosis, maintenance, qualified, manufacturer specific diagnosis, or manufacturer specific status information. It consists of the following attributes:

IP Address

This attribute shall be set to the IO device's IP address.

Attribute type: Unsigned32

Name Of Station Length

This attribute shall be set to the length of the IO device's Name Of Station.

Attribute type: Unsigned16

Name Of Station Value

This attribute shall be set to the IO device's Name Of Station.

Attribute type: VisibleString[240]

7.3.4.2.3 Invocation of the Diagnosis object

An IO device shall invoke one Diagnosis object for each application process which shall contain the diagnosis information of the IO device with its faulty or maintenance information containing slots, subslots and channels.

7.3.4.3 Diagnosis service specification

7.3.4.3.1 Add Diagnosis Entry

This service is used to add a diagnosis entry to the diagnosis ASE. Table 140 shows the parameters of the service.

Table 140 – Local Add Diagnosis Entry

Parameter name	req	cnf
Argument	M	
AREP	M	
API	M	
Slot Number	M	
Subslot Number	M	
List of Channel Diagnosis Data	M	
Channel Number	M	
Channel Properties	M	
User Structure Identifier	M	
Channel Error type	S	

Parameter name	req	cnf
Ext Channel Error type	S	
Ext Channel Add Value	S	
Qualified Channel Qualifier	S	
List of Manufacturer Specific Diagnosis Data	S	
Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR.

API

This parameter contains the value of the corresponding attribute of the ASE object.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

List of Channel Diagnosis Data

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

User Structure Identifier

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Error type

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Error type

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Add Value

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Qualified Channel Qualifier

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to QUALIFIED_CHANNEL_DIAGNOSIS_USI.

List of Manufacturer Specific Diagnosis Data

This optional parameter is composed of the following elements.

It is present only, if the attribute User Structure Identifier is set in the range from MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF.

Data

This parameter contains the value of the corresponding attribute of the ASE object.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.4.3.2 Remove Diagnosis Entry

This service is used to remove a diagnosis entry from the diagnosis ASE. Table 141 shows the parameters of the service.

Table 141 – Local Remove Diagnosis Entry

Parameter name	req	cnf
Argument	M	
AREP	M	
API	M	
Slot Number	M	
Subslot Number	M	
List of Channel Diagnosis Data	M	
Channel Number	M	
Channel Properties	M	
User Structure Identifier	M	
Channel Error type	S	
Ext Channel Error type	S	
Ext Channel Add Value	S	
Qualified Channel Qualifier	S	
List of Manufacturer Specific Diagnosis Data	S	
Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

The argument shall convey the service specific parameters of the service indication.

AREP

This parameter is the local identifier for the desired AR.

API

This parameter contains the value of the corresponding attribute of the ASE object.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

List of Channel Diagnosis Data

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

User Structure Identifier

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Error type

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Error type

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Add Value

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Qualified Channel Qualifier

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to QUALIFIED_CHANNEL_DIAGNOSIS_USI.

List of Manufacturer Specific Diagnosis Data

This optional parameter is composed of the following elements.

It is present only, if the attribute User Structure Identifier is set in the range from MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF.

Data

This parameter contains the value of the corresponding attribute of the ASE object.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

7.3.4.3.3 Local Diagnosis Event

This service is used to indicate a diagnosis event detected by protocol machines. Table 142 shows the parameters of the service.

Furthermore, the attribute values of the Diagnosis ASE shall be set according to the service parameter. Additionally, an Alarm service shall be issued if it is required (e.g. for Alarm type "Multicast Provider Communication Stopped").

Table 142 – Local Diagnosis Event

Parameter name	ind
Argument	M
API	M
Slot Number	M
Subslot Number	M
List of Channel Diagnosis Data	M
Channel Number	M
Channel Properties	M
User Structure Identifier	M
Channel Error type	S
Ext Channel Error type	S
Ext Channel Add Value	S
Qualified Channel Qualifier	S
List of Manufacturer Specific Diagnosis Data	S
Data	M

Argument

The argument shall convey the service specific parameters of the service indication.

API

This parameter contains the value of the corresponding attribute of the ASE object.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

List of Channel Diagnosis Data

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

User Structure Identifier

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Error type

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Error type

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Add Value

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI or QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Qualified Channel Qualifier

This optional parameter contains the value of the corresponding attribute of the ASE object.

It is present only, if the attribute User Structure Identifier is set to QUALIFIED_CHANNEL_DIAGNOSIS_USI.

List of Manufacturer Specific Diagnosis Data

This optional parameter is composed of the following elements.

It is present only, if the attribute User Structure Identifier is set in the range from MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF.

Data

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.4.3.4 General Data Definition

All services to read the diagnosis object have the same parameter layout. To ease the description the parameter is thus described once upfront to the services. The services itself act as a filter for the information from the diagnosis ASE.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies.

The parameter Data has a service specific structure and meaning. Table 143 shows the specific elements of Data for this service. Note that dependent on the particular service described below only parts of the information are returned (e.g. no manufacturer specific diagnosis data...).

Table 143 – General Data definition for Diagnosis services

Parameter name	service specific elements	rsp	cnf
Data	List of Diagnosis Data	M	M(=)

Parameter name	service specific elements	rsp	cnf
	API	M	M(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	User Structure Identifier	M	M(=)
	List of Channel Diagnostic Data	S	S(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	Channel Error type	M	M(=)
	List of Ext Channel Diagnosis Data	S	S(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	Channel Error type	M	M(=)
	Ext Channel Error type	M	M(=)
	Ext Channel Add Value	M	M(=)
	List of Qualified Ext Channel Diagnosis Data	M	M(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	Channel Error type	M	M(=)
	Ext Channel Error type	M	M(=)
	Ext Channel Add Value	M	M(=)
	Qualified Channel Qualifier	M	M(=)
	Manufacturer Specific Diagnosis	S	S(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	List of Data	M	M(=)
	Data	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of Diagnosis Data

This parameter contains a list of diagnosis data. A list element consists of the following parameters:

API

This parameter contains the value of the corresponding attribute of the ASE object.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Number

This parameter shall be set to 0x8000.

Channel Properties

The field ChannelProperties.Type, the field ChannelProperties.Direction, the field ChannelProperties.Maintenance shall be set to zero.

The field ChannelProperties.Specifier shall be set to APPEAR if at least one ChannelProperties.Specifier in the diagnosis data is set to APPEAR in conjunction with ChannelProperties.Maintenance(=diagnosis). Else, the field ChannelProperties.Specifier shall be set to DISAPPEAR.

User Structure Identifier

This parameter indicates the format of the following data.

Attribute type: Unsigned16

Allowed values: CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, QUALIFIED_CHANNEL_DIAGNOSIS_USI, MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF

NOTE The IO device determines in which format a specific diagnosis item is returned. The requestor is able to determine, whether manufacturer specific data shall be returned (by requesting "all codings") or not (by requesting "channel coding").

List of Channel Diagnostic Data

This optional parameter contains a list of channel diagnosis data related to this channel. It shall be present only if the parameter User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI.

A list element consists of the following parameters:

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Error type

This parameter contains the value of the corresponding attribute of the ASE object.

List of Ext Channel Diagnostic Data

This optional parameter contains a list of extended channel diagnosis data related to this channel. It shall be present only if the parameter User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI.

A list element consists of the following parameters:

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Error type

This parameter contains the value of the corresponding attribute of the ASE object.

Ext Channel Error type

This parameter contains the value of the corresponding attribute of the ASE object.

Ext Channel Add Value

This parameter contains the value of the corresponding attribute of the ASE object.

Qualified Ext Channel Diagnosis

This optional parameter contains a list of extended channel diagnosis data related to this channel. It shall be present only if the parameter User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI.

A list element consists of the following parameters:

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Error type

This parameter contains the value of the corresponding attribute of the ASE object.

Ext Channel Error type

This parameter contains the value of the corresponding attribute of the ASE object.

Ext Channel Add Value

This parameter contains the value of the corresponding attribute of the ASE object.

Qualified Channel Qualifier

This parameter contains the value of the corresponding attribute of the ASE object.

Manufacturer Specific Diagnosis

This optional parameter contains a list of extended channel diagnosis data related to this channel. It shall be present only if the parameter User Structure Identifier is set to MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF.

A list element consists of the following parameters:

Channel Number

This parameter contains the value of the corresponding attribute of the ASE object.

Channel Properties

This parameter contains the value of the corresponding attribute of the ASE object.

List of Data

This parameter contains a list of manufacturer specific diagnosis data. A list element consists of the following parameter.

Data

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.4.3.5 Read Diagnosis in channel coding for one subslot

This confirmed service may be used to read diagnosis data in channel coding for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in channel coding for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.6 Read Diagnosis in all codings for one subslot

This confirmed service may be used to read diagnosis data in all codings for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in all codings for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.7 Read Maintenance required in channel coding for one subslot

This confirmed service may be used to read maintenance data in channel coding for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in channel coding for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.8 Read Maintenance required in all codings for one subslot

This confirmed service may be used to read maintenance data in all codings for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in all codings for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.9 Read Maintenance demanded in channel coding for one subslot

This confirmed service may be used to read maintenance data in channel coding for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in channel coding for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.10 Read Maintenance demanded in all codings for one subslot

This confirmed service may be used to read maintenance data in all codings for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in all codings for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.11 Read Diagnosis, Maintenance, Qualified and Status for one subslot

This confirmed service may be used to read maintenance data in all codings for one subslot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis, maintenance, qualified and status for one subslot belongs to a submodule, which is addressed by its API, slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed submodule, and returns information for those channels, where

- ChannelProperties.Maintenance is set to ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) and ChannelProperties.Specifier is set to APPEAR, or
- ChannelProperties.Maintenance is set to DIAGNOSIS, ChannelProperties.Specifier is set to DISAPPEAR, and manufacturer specific diagnosis data exists (i.e. status information).

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.12 Read Diagnosis in channel coding for one slot

This confirmed service may be used to read diagnosis data in channel coding for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in channel coding for one slot belongs to a module, which is addressed by its API and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed module, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.13 Read Diagnosis in all codings for one slot

This confirmed service may be used to read diagnosis data in all codings for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in all codings for one slot belongs to a module, which is addressed by its API, and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed module, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.14 Read Maintenance required in channel coding for one slot

This confirmed service may be used to read maintenance data in channel coding for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in channel coding for one subslot belongs to a module, which is addressed by its API and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed module, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.15 Read Maintenance required in all codings for one slot

This confirmed service may be used to read maintenance data in all codings for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5

applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in all codings for one subslot belongs to a module, which is addressed by its API and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed module, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.16 Read Maintenance demanded in channel coding for one slot

This confirmed service may be used to read maintenance data in channel coding for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in channel coding for one subslot belongs to a module, which is addressed by its API and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed submodule, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.17 Read Maintenance demanded in all codings for one slot

This confirmed service may be used to read maintenance data in all codings for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in all codings for one subslot belongs to a module, which is addressed by its API and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to submodules of the addressed module, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.18 Read Diagnosis, Maintenance, Qualified and Status for one slot

This confirmed service may be used to read maintenance data in all codings for one slot. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning, see 7.3.4.3.4 for its general description.

Diagnosis, maintenance, qualified and status for one slot belongs to a module, which is addressed by its API and slot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to submodules of the addressed module, and returns information for those channels, where

- ChannelProperties.Maintenance is set to ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) and ChannelProperties.Specifier is set to APPEAR, or
- ChannelProperties.Maintenance is set to DIAGNOSIS, ChannelProperties.Specifier is set to DISAPPEAR, and manufacturer specific diagnosis data exists (i.e. status information).

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.19 Read Diagnosis in channel coding for one AR

This confirmed service may be used to read diagnosis data in channel coding for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in channel coding for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules within the addressed AR, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.20 Read Diagnosis in all codings for one AR

This confirmed service may be used to read diagnosis data in all codings for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in all codings for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules within the addressed AR, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.21 Read Maintenance required in channel coding for one AR

This confirmed service may be used to read maintenance data in channel coding for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in channel coding for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed AR, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.22 Read Maintenance required in all codings for one AR

This confirmed service may be used to read maintenance data in all codings for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in all codings for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed AR, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.23 Read Maintenance demanded in channel coding for one AR

This confirmed service may be used to read maintenance data in channel coding for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in channel coding for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed AR, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.24 Read Maintenance demanded in all codings for one AR

This confirmed service may be used to read maintenance data in all codings for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in all codings for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to submodules of the addressed AR, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.25 Read Diagnosis, Maintenance, Qualified and Status for one AR

This confirmed service may be used to read maintenance data in all codings for one AR. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning, see 7.3.4.3.4 for its general description.

Diagnosis, maintenance, qualified and status for one AR belongs to an application relation, which is addressed by ARUUID or if using an implicit AR by Target AR UUID. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to submodules of the addressed AR, and returns information for those channels, where

- ChannelProperties.Maintenance is set to ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) and ChannelProperties.Specifier is set to APPEAR, or
- ChannelProperties.Maintenance is set to DIAGNOSIS, ChannelProperties.Specifier is set to DISAPPEAR, and manufacturer specific diagnosis data exists (i.e. status information).

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.26 Read Diagnosis in channel coding for one API

This confirmed service may be used to read diagnosis data in channel coding for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in channel coding for one API belongs to a AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed API, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.27 Read Diagnosis in all codings for one API

This confirmed service may be used to read diagnosis data in all codings for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis in all codings for one API belongs to a AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to the addressed API, and returns information for those channels, where ChannelProperties.Maintenance is set to DIAGNOSIS and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.28 Read Maintenance required in channel coding for one API

This confirmed service may be used to read maintenance data in channel coding for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in channel coding for one API belongs to a AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed API, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.29 Read Maintenance required in all codings for one API

This confirmed service may be used to read maintenance data in all codings for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance required in all codings for one subslot belongs to an AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed API, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_REQUIRED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.30 Read Maintenance demanded in channel coding for one API

This confirmed service may be used to read maintenance data in channel coding for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in channel coding for one API belongs to a AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed API, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Only channel diagnostic data, extended channel diagnostic data and qualified diagnostic data is returned; or in other words manufacturer specific diagnostic data is not returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned.

7.3.4.3.31 Read Maintenance demanded in all codings for one API

This confirmed service may be used to read maintenance data in all codings for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Maintenance demanded in all codings for one API belongs to a AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to submodules of the addressed API, and returns information for those channels, where ChannelProperties.Maintenance is set to MAINTENANCE_DEMANDED and ChannelProperties.Specifier is set to APPEAR.

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.32 Read Diagnosis, Maintenance, Qualified and Status for one API

This confirmed service may be used to read maintenance data in all codings for one API. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning, see 7.3.4.3.4 for its general description.

Diagnosis, maintenance, qualified and status for one API belongs to a AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to submodules of the addressed module, and returns information for those channels, where

- ChannelProperties.Maintenance is set to ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) and ChannelProperties.Specifier is set to APPEAR, or
- ChannelProperties.Maintenance is set to DIAGNOSIS, ChannelProperties.Specifier is set to DISAPPEAR, and manufacturer specific diagnosis data exists (i.e. status information).

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.33 Read Diagnosis, Maintenance, Qualified and Status for one device

This confirmed service may be used to read maintenance data in all codings for one device. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

Diagnosis, maintenance, qualified and status for one device belongs to the device. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the diagnosis object regarding all channels belonging to all submodules of the addressed device, and returns information for those channels, where

- ChannelProperties.Maintenance is set to ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) and ChannelProperties.Specifier is set to APPEAR, or
- ChannelProperties.Maintenance is set to DIAGNOSIS, ChannelProperties.Specifier is set to DISAPPEAR, and manufacturer specific diagnosis data exists (i.e. status information).

Channel diagnostic data, extended channel diagnostic data, qualified diagnostic data, and manufacturer specific diagnostic data are returned.

NOTE The IO device determines in which format (channel diagnostic data, extended channel diagnostic data or qualified diagnostic data) a specific diagnosis item is returned. Manufacturer specific diagnosis however is returned always in the manufacturer specific diagnostic data format.

7.3.4.3.34 Read Nested Diagnosis Information

This optional service may be used to read the nested diagnosis information. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning. Table 144 shows the specific elements of Data for this service.

Nested diagnosis information belongs to the device. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

Table 144 – Data elements of Read Nested Diagnosis Information

Parameter name	service specific elements	rsp	cnf
Data	Number of Entries	M	M(=)
	List of Nested Diagnosis Info	M	M(=)
	Slot Number	M	M(=)
	IP Address	M	M(=)
	Station Name Length	M	M(=)
	Name Of Station Value	M	M(=)

Data

This parameter contains the values of the object which has been read.

Number Of Entries

This parameter indicates the number of entries in the following list.

Attribute type: Unsigned16

List of Nested Diagnosis Info

This parameter is composed of the following list elements:

IP Address

This parameter contains the value of the corresponding attribute of the ASE object.

Station Name Length

This parameter contains the value of the corresponding attribute of the ASE object.

Name Of Station Value

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.4.4 Behavior of Diagnosis objects

7.3.4.4.1 General behavior of the Diagnosis object

An entry of a Diagnosis object shall maintain different states for diagnosis, maintenance required, maintenance demanded, and qualified 2 to 32.

7.3.4.4.2 Behavior Diagnosis entry

7.3.4.4.2.1 Primitive definitions

7.3.4.4.2.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by diagnosis entry state machine (DIAG_DIAG) are described in the service definition and shown in Table 145.

Table 145 – Remote primitives issued or received by DIAG_DIAG

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	DIAG_DIAG	FAL	see 7.3.5.3	—

7.3.4.4.2.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by DIAG_DIAG are described in the service definition and shown in Table 146.

Table 146 – Local primitives issued or received by DIAG_DIAG

Primitive	Source	Destination	Associated parameters	Functions
AddDiagnosisEntry_req ()	FAL	DIAG_DIAG	see service specification	—
AddDiagnosisEntry_cnf ()	DIAG_DIAG	FAL	see service specification	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_DIAG	see service specification	—
RemoveDiagnosisEntry_cnf ()	DIAG_DIAG	FAL	see service specification	—

7.3.4.4.2.2 State transition diagram

The state transition diagram is shown in Figure 28.

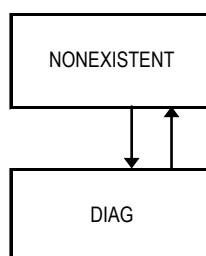


Figure 28 – State transition diagram DIAG_DIAG

States of the DIAG_DIAG

NONEXISTENT

This state indicate that no diagnosis entry exists or an existing entry is removed and therefore an alarm request is sent

DIAG

This state indicate that a diagnosis entry exists and an alarm request is sent

7.3.4.4.2.3 State machine description

This machine is intended to show that the diagnosis behavior consist of two steps for the user. The first step is used for register the diagnosis data to the diagnosis ASE and the second step sends an alarm request with the diagnosis data.

If the diagnosis disappears, also two steps are necessary. The local data must be updated and the controller must be informed during an alarm request.

To simplify the state machine the confirmation is not included.

7.3.4.4.2.4 State table

Table 147 defines the state table DIAG_DIAG.

Table 147 – State table DIAG_DIAG

#	Current State	Event /Condition =>Action	Next State
1	NONEXISTENT	AddDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType= (CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => CreateDiagnosisEntry () UpdateAPDUStatus (For all related ARs) Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	DIAG
2	DIAG	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => RemoveDiagnosisEntry () UpdateAPDUStatus (For all related ARs) Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.2.5 Functions, Macros, Timers and Variables

Table 148 contains the functions, macros, timers and variables used by DIAG_DIAG and their arguments and their descriptions.

Table 148 – Functions, Macros, Timers and Variables used by DIAG_DIAG

Name	Type	Function/Meaning
BuildDiagnosisAlarmAppears	Function	This local function returns the diagnosis entry of the given format for the alarm notification (Channel Properties. Specifier := APPEAR) using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
BuildDiagnosisAlarmDisappears	Function	This local function returns the diagnosis entry of the given format for the alarm notification, the diagnosis entry (Channel Properties. Specifier) shall contain the summarized diagnosis info for the whole API/module/submodule/channel to select DISAPPEAR or DISAPPEAR_OTHER_EXIST using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Function	This local function returns the alarm specifier entry for the alarm notification, the alarm specifier shall contain the summarized alarm info for the whole API/module/submodule (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) and for all API/modules/submodules of the related AR (AR Diagnosis State) using the following parameters: API, Slot, Subslot
BuildSummarizedMaintenanceStatus	Function	This local function returns the Maintenance Status entry for the alarm notification, it shall contain the summarized maintenance information of the API/module/submodule (required, demanded, or qualified 2 to 31) if no maintenance is present nothing shall be returned and the optional service parameter shall be omitted in the alarm notification using the following parameters: API, Slot, Subslot
CreateDiagnosisEntry	Function	This local function invokes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Function	This local function returns the alarm priority for the alarm type dedicated to a certain AR using the following parameter: AREP

Name	Type	Function/Meaning
DeleteDiagnosisEntry	Function	This local function deletes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
ApplicationDiagnosisDetected	Macro	This event is called by the application process of an IO device to indicate a diagnosis using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	This event is called by the application process of an IO device to indicate that a diagnosis is no longer existing using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType
UpdateAPDUStatus	Macro	This macro sets the Problem Indicator Flag to TRUE for all related ARs if at least one diagnosis appear entry for the API/module/submodule exist, otherwise it shall be set to FALSE

7.3.4.4.3 Behavior maintenance required entry

7.3.4.4.3.1 Primitive definitions

7.3.4.4.3.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by maintenance required entry state machine (DIAG_MR) are described in the service definition and shown in Table 149.

Table 149 – Remote primitives issued or received by DIAG_MR

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	DIAG_MR	FAL	see 7.3.5.3	—

7.3.4.4.3.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by DIAG_MR are described in the service definition and shown in Table 150.

Table 150 – Local primitives issued or received by DIAG_MR

Primitive	Source	Destination	Associated parameters	Functions
AddDiagnosisEntry_req ()	FAL	DIAG_MR	see service specification	—

AddDiagnosisEntry_cnf ()	DIAG_MR	FAL	see service specification	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_MR	see service specification	—
RemoveDiagnosisEntry_cnf ()	DIAG_MR	FAL	see service specification	—

7.3.4.4.3.2 State transition diagram

The state transition diagram is shown in Figure 29.

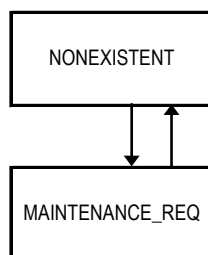


Figure 29 – State transition diagram DIAG_MR

States of the DIAG_MR

NONEXISTENT

This state indicate that no diagnosis entry exists or an existing entry is removed and therefore an alarm request is sent

MAINTENANCE_REQ

This state indicate that a diagnosis entry exists and an alarm request is sent

7.3.4.4.3.3 State machine description

This machine is intended to show that the diagnosis behavior consist of two steps for the user. The first step is used for register the maintenance required diagnosis data to the diagnosis ASE and the second step sends an alarm request with the diagnosis data.

If the diagnosis disappears, also two steps are necessary. The local data must be updated and the controller must be informed during an alarm request.

To simplify the state machine the confirmation is not included.

7.3.4.4.3.4 State table

Table 151 defines the state table DIAG_MR.

Table 151 – State table DIAG_MR

#	Current State	Event /Condition =>Action	Next State

1	NONEXISTENT	AddDiagnosisEntry_req () /ChannelProperties.Maintenance Required=TRUE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType= (CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => CreateDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	MAINTENANCE_REQ
2	MAINTENANCE_REQ	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=TRUE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => DeleteDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.3.5 Functions, Macros, Timers and Variables

Table 152 contains the functions, macros, timers and variables used by DIAG_MR and their arguments and their descriptions.

Table 152 – Functions, Macros, Timers and Variables used by DIAG_MR

Name	Type	Function/Meaning
BuildDiagnosisAlarmAppears	Function	This local function returns the diagnosis entry of the given format for the alarm notification (Channel Properties. Specifier := APPEAR) using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
BuildDiagnosisAlarmDisappears	Function	This local function returns the diagnosis entry of the given format for the alarm notification, the diagnosis entry (Channel Properties. Specifier) shall contain the summarized diagnosis info for the whole API/module/submodule/channel to select DISAPPEAR or DISAPPEAR_OTHER_EXIST using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Function	This local function returns the alarm specifier entry for the alarm notification, the alarm specifier shall contain the summarized alarm info for the whole API/module/submodule (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) and for all API/modules/submodules of the related AR (AR Diagnosis State) using the following parameters: API, Slot, Subslot
BuildSummarizedMaintenanceStatus	Function	This local function returns the Maintenance Status entry for the alarm notification, it shall contain the summarized maintenance information of the API/module/submodule (required, demanded, or qualified 2 to 31) if no maintenance is present nothing shall be returned and the optional service parameter shall be omitted in the alarm notification using the following parameters: API, Slot, Subslot
CreateDiagnosisEntry	Function	This local function invokes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Function	This local function returns the alarm priority for the alarm type dedicated to a certain AR using the following parameter: AREP
DeleteDiagnosisEntry	Function	This local function deletes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties

Name	Type	Function/Meaning
ApplicationDiagnosisDetected	Macro	This event is called by the application process of an IO device to indicate a diagnosis using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	This event is called by the application process of an IO device to indicate that a diagnosis is no longer existing using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType

7.3.4.4.4 Behavior maintenance demanded entry

7.3.4.4.4.1 Primitive definitions

7.3.4.4.4.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by maintenance demanded entry state machine (DIAG_MD) are described in the service definition and shown in Table 153.

Table 153 – Remote primitives issued or received by DIAG_MD

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	DIAG_MD	FAL	see 7.3.5.3	—

7.3.4.4.4.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by DIAG_MD are described in the service definition and shown in Table 154.

Table 154 – Local primitives issued or received by DIAG_MD

Primitive	Source	Destination	Associated parameters	Functions
AddDiagnosisEntry_req ()	FAL	DIAG_MD	see service specification	—
AddDiagnosisEntry_cnf ()	DIAG_MD	FAL	see service specification	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_MD	see service specification	—
RemoveDiagnosisEntry_cnf ()	DIAG_MD	FAL	see service specification	—

7.3.4.4.4.2 State transition diagram

The state transition diagram is shown in Figure 30.

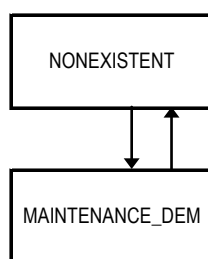


Figure 30 – State transition diagram DIAG_MD

States of the DIAG_MD

NONEXISTENT

This state indicate that no diagnosis entry exists or an existing entry is removed and therefore an alarm request is sent

MAINTENANCE_DEM

This state indicate that a diagnosis entry exists and an alarm request is sent

7.3.4.4.4.3 State machine description

This machine is intended to show that the diagnosis behavior consist of two steps for the user. The first step is used for register the maintenance demanded diagnosis data to the diagnosis ASE and the second step sends an alarm request with the diagnosis data.

If the diagnosis disappears, also two steps are necessary. The local data must be updated and the controller must be informed during an alarm request.

To simplify the state machine the confirmation is not included.

7.3.4.4.4.4 State table

Table 155 defines the state table DIAG_MD.

Table 155 – State table DIAG_MD

#	Current State	Event /Condition =>Action	Next State
1	NONEXISTENT	AddDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType= (CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => CreateDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	MAINTENANCE_DEM

2	MAINTENANCE_DEM	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => DeleteDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.5 Functions, Macros, Timers and Variables

Table 156 contains the functions, macros, timers and variables used by DIAG_MD and their arguments and their descriptions.

Table 156 – Functions, Macros, Timers and Variables used by the maintenance demanded entry

Name	Type	Function/Meaning
BuildDiagnosisAlarmAppears	Function	This local function returns the diagnosis entry of the given format for the alarm notification (Channel Properties. Specifier := APPEAR) using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
BuildDiagnosisAlarmDisappears	Function	This local function returns the diagnosis entry of the given format for the alarm notification, the diagnosis entry (Channel Properties. Specifier) shall contain the summarized diagnosis info for the whole API/module/submodule/channel to select DISAPPEAR or DISAPPEAR_OTHER_EXIST using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Function	This local function returns the alarm specifier entry for the alarm notification, the alarm specifier shall contain the summarized alarm info for the whole API/module/submodule (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) and for all API/modules/submodules of the related AR (AR Diagnosis State) using the following parameters: API, Slot, Subslot

Name	Type	Function/Meaning
BuildSummarizedMaintenanceStatus	Function	This local function returns the Maintenance Status entry for the alarm notification, it shall contain the summarized maintenance information of the API/module/submodule (required, demanded, or qualified 2 to 31) if no maintenance is present nothing shall be returned and the optional service parameter shall be omitted in the alarm notification using the following parameters: API, Slot, Subslot
CreateDiagnosisEntry	Function	This local function invokes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Function	This local function returns the alarm priority for the alarm type dedicated to a certain AR using the following parameter: AREP
DeleteDiagnosisEntry	Function	This local function deletes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
ApplicationDiagnosisDetected	Macro	This event is called by the application process of an IO device to indicate a diagnosis using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	This event is called by the application process of an IO device to indicate that a diagnosis is no longer existing using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType

7.3.4.4.5 Behavior qualified entry

7.3.4.4.5.1 Primitive definitions

7.3.4.4.5.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by qualified entry state machine (DIAG_QUALIFIED) are described in the service definition and shown in Table 157.

Table 157 – Remote primitives issued or received by DIAG_QUALIFIED

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	DIAG_QUALIFIED	FAL	see 7.3.5.3	—

7.3.4.4.5.1.2 Primitives exchanged between local machines

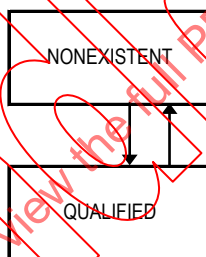
The local service primitives including their associated parameters issued or received by DIAG_QUALIFIED are described in the service definition and shown in Table 158.

Table 158 – Local primitives issued or received by DIAG_QUALIFIED

Primitive	Source	Destination	Associated parameters	Functions
AddDiagnosisEntry_req ()	FAL	DIAG_QUALIFIED	see service specification	—
AddDiagnosisEntry_cnf ()	DIAG_QUALIFIED	FAL	see service specification	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_QUALIFIED	see service specification	—
RemoveDiagnosisEntry_cnf ()	DIAG_QUALIFIED	FAL	see service specification	—

7.3.4.4.5.2 State transition diagram

The state transition diagram is shown in Figure 31.

**Figure 31 – State transition diagram DIAG_QUALIFIED**

States of the DIAG_QUALIFIED

NONEXISTENT

This state indicate that no diagnosis entry exists or an existing entry is removed and therefore an alarm request is sent

QUALIFIED

This state indicate that a diagnosis entry exists and an alarm request is sent

7.3.4.4.5.3 State machine description

This machine is intended to show that the diagnosis behavior consist of two steps for the user. The first step is used for register the diagnosis data to the diagnosis ASE and the second step sends an alarm request with the diagnosis data.

If the diagnosis disappears, also two steps are necessary. The local data must be updated and the controller must be informed during an alarm request.

To simplify the state machine the confirmation is not included.

7.3.4.4.5.4 State table

Table 159 defines the state table which exists for each qualified entry (2 to 31).

Table 159 – State table DIAG_QUALIFIED

#	Current State	Event /Condition =>Action	Next State
1	NONEXISTENT	AddDiagnosisEntry_req (Data=NIL) /ChannelProperties.Maintenance Required=TRUE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType= (QUALIFIED_EXT_CHANNELDIAGNOSIS) => CreateDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears (Data=NIL) Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	QUALIFIED
2	QUALIFIED	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => DeleteDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.5.5 Functions, Macros, Timers and Variables

Table 160 contains the functions, macros, timers and variables used by DIAG_QUALIFIED and their arguments and their descriptions.

Table 160 – Functions, Macros, Timers and Variables used by DIAG_QUALIFIED

Name	Type	Function/Meaning
BuildDiagnosisAlarmAppears	Function	This local function returns the diagnosis entry of the given format for the alarm notification (Channel Properties. Specifier := APPEAR) using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
BuildDiagnosisAlarmDisappears	Function	This local function returns the diagnosis entry of the given format for the alarm notification, the diagnosis entry (Channel Properties. Specifier) shall contain the summarized diagnosis info for the whole API/module/submodule/channel to select DISAPPEAR or DISAPPEAR_OTHER_EXIST using the following parameters: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType

Name	Type	Function/Meaning
BuildSummarizedAlarmSpecifier	Function	This local function returns the alarm specifier entry for the alarm notification, the alarm specifier shall contain the summarized alarm info for the whole API/module/submodule (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) and for all API/modules/submodules of the related AR (AR Diagnosis State) using the following parameters: API, Slot, Subslot
BuildSummarizedMaintenanceStatus	Function	This local function returns the Maintenance Status entry for the alarm notification, it shall contain the summarized maintenance information of the API/module/submodule (required, demanded, or qualified 2 to 31) if no maintenance is present nothing shall be returned and the optional service parameter shall be omitted in the alarm notification using the following parameters: API, Slot, Subslot
CreateDiagnosisEntry	Function	This local function invokes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Function	This local function returns the alarm priority for the alarm type dedicated to a certain AR using the following parameter: AREP
DeleteDiagnosisEntry	Function	This local function deletes a diagnosis entry of the given type within the diagnosis object using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
ApplicationDiagnosisDetected	Macro	This event is called by the application process of an IO device to indicate a diagnosis using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	This event is called by the application process of an IO device to indicate that a diagnosis is no longer existing using the following parameters: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType, QualifiedChannelQualifier

7.3.4.4.6 Optimization for Diagnosis objects

7.3.4.4.6.1 Common

A diagnosis object shall be fully addressed with API, Slot Number, Subslot Number, Channel Number, Channel Error type, Ext Channel Error type, and Channel Properties (using only Maintenance required, Maintenance demanded and Specifier).

7.3.4.4.6.2 Subslot Number level

If all diagnosis objects of the address level subslot (one Subslot Number) are deleted at once, then a Diagnosis disappears alarm without an alarm item shall be generated.

This signals, that all subsequent diagnosis, maintenance (required and demanded) and qualified are gone.

7.3.4.4.6.3 Channel Error type level

If all diagnosis objects of the address level Channel Error type (one Channel Error type) are deleted at once, then an alarm item with the following content shall be generated:

- Channel Number
- Channel Properties
 - Type
 - Direction
 - Maintenance Required and Maintenance Demanded shall be zero
 - Specifier shall be zero
- Channel Error type

This signals, that all subsequent diagnosis, maintenance (required and demanded) and qualified are gone.

NOTE Subsequent in this case means, also all subsequent diagnosis objects addressed by Ext Channel Error Types are gone.

7.3.5 Alarm ASE

7.3.5.1 Overview

The alarm model allows the transfer of an alarm from the IO device to the assigned IO controller or vice versa and the explicit acknowledgement of the alarm.

For the transmission of an alarm the following conditions have to be fulfilled:

- the IO AR shall be established
- the source submodule shall be owned by the IO AR
- the source submodule shall be existent (no empty subslot and no wrong submodule).

An alarm shall always be issued; the sink shall respond with a negative acknowledgment “not supported” in case no handler for the Alarm type is installed.

Table 161 gives an overview over the defined alarm types, their usage and whether the support of the alarm is mandatory or optional.

The alarm types indicated as “attached to Diagnosis ASE” are issued, whenever entries are added or removed from the Diagnosis ASE. Those alarms thus allow maintaining the actual diagnosis state of an IO device. The services of the diagnosis ASE allow getting the current state of the diagnosis.

Note that the alarms attached to the diagnosis ASE may convey for of a channel several diagnosis items (to reduce the number of alarms to be sent)

The columns “alarm payload” define the items being part of the alarm payload of a particular alarm.

Table 161 – Alarm type

Alarm type	Usage	Use	Attached to Diagnosis ASE	Alarm Payload				
				Maintenance Item	Diagnosis Item	Upload & Retrieval iParameter Item	Manufacturer Alarm Item	Profile Alarm Item
Diagnosis	The value Diagnosis shall be used to indicate appearing or disappearing items for one or more channels	Mandatory	Yes	a	b	—	—	—
Process	Indicates an alarm from the controlled process (usage application and process specific)	Optional	No	—	—	—	c	—
Pull	Shall indicate an alarm if a used module/submodule is pulled	Mandatory	No	—	—	—	c	—
Plug	Shall indicate an alarm if a requested submodule is plugged	Mandatory	No	—	—	—	c	—
Status	Indicates a status change within a submodule	Optional	No	—	—	—	c	—
Update	Indicates a change of parameter for a submodule	Optional	No	—	—	—	c	—
Media Redundancy	Indicates a change in the media redundancy status	Mandatory when supporting media redundancy	Yes	a	b	—	—	—
Controlled by supervisor	Shall indicate an alarm if a used submodule has been locked by an IO supervisor Logical “pull” of a submodule to withdraw ownership	Mandatory	No	—	—	—	c	—
Released	Shall indicate an alarm if a requested submodule has been unlocked by an IO supervisor, IO Controller or by IO Device local means Logical “plug” of a submodule to return ownership or trigger a reparameterization	Mandatory	No	—	—	—	c	—
Plug Wrong Submodule	Shall indicate an alarm if a wrong submodule for a requested module/submodule is plugged	Mandatory	No	—	—	—	c	—
Return Of Submodule	Shall indicate a change of IOCS/IOPS from BAD to GOOD for a submodule	Mandatory	No	—	—	—	c	—

Alarm type	Usage	Use	Attached to Diagnosis ASE	Alarm Payload				
				Maintenance Item	Diagnosis Item	Upload & Retrieval iParameter Item	Manufacturer Alarm Item	Profile Alarm Item
Diagnosis Disappears	The value Diagnosis Disappears shall only be used - to indicate all disappearing without an Alarm Item (optimization) - to indicate detailed disappearing items	Mandatory	Yes	—	—	—	d	—
Multicast Communication Mismatch Notification	A multicast consumer shall indicate that the multicast CR has had a problem	Mandatory when supporting multicast	Yes	a	b	—	—	—
Port Data Change Notification	Indicates a change of port data e.g. link up or down	Mandatory	Yes	a	b	—	—	—
Sync Data Changed Notification	Indicates a change of clock synchronization	Mandatory when supporting clock synchronization	Yes	a	b	—	—	—
Isochronous Mode Problem Notification	Indicates a problem of isochronous applications, e.g. started to late	Mandatory when supporting isochronous mode	Yes	a	b	—	—	—
Network Component Problem Notification	Indicates a problem of network components, e.g. frame dropped no resources	Mandatory when supporting network component diagnosis	Yes	a	b	—	—	—
Time Data Changed Notification	Indicates a change of time synchronization	Mandatory when supporting time synchronization	Yes	a	b	—	—	—
Dynamic Frame Packing problem Notification	Indicates a problem during the dynamic frame packing operation	Mandatory when supporting dynamic frame packing	Yes	a	b	—	—	—
MRPD problem Notification	Indicates a problem during MRPD operation	Optional when supporting MRPD	Yes	a	b	—	—	—
System Redundancy	Indicates an IO controller that it holds no longer the primary AR	Mandatory when supporting IO controller redundancy	Yes	a	b	—	—	—
Multiple Interface Mismatch Notification	Indicates a inconsistent setting of parameters affecting more than one interface	Mandatory	Yes	a	b	—	—	—
Upload and Retrieval Notification	Indicates a request for upload and storage of Record Data objects	Optional	No	—	—	e	—	—
Pull module	Shall indicate an alarm if a used module is pulled	Mandatory when supporting the pull module alarm	No	—	—	—	c	—
Manufacturer Specific	Values from 0x0020 to 0x007F may be used for manufacturer specific alarm types	Optional	No	—	—	—	c	—

Alarm type	Usage	Use	Attached to Diagnosis ASE	Alarm Payload				
				Maintenance Item	Diagnosis Item	Upload & Retrieval IParameter Item	Manufacturer Alarm Item	Profile Alarm Item
Profile Specific	Shall be used according Profile Guidelines if an IO device complies to a specific device profile	According to supported profile	No	—	—	—	—	f
a The maintenance item is conveyed, if the diagnosis item contains at least one channel diagnosis (including extended, qualified or manufacturer specific diagnosis) with a ChannelProperties Maintenance set to MAINTENANCE_REQUIRED or MAINTENANCE_DEMANDED. b The diagnosis item can be omitted if all diagnosis disappear from the channel. c The alarm information may contain optionally manufacturer specific information. d The alarm information may contain optionally manufacturer specific information; however this is not recommended. e The alarm payload is extended with either an Upload and Retrieval Item or else an IParameter Item. f The alarm information may contain optionally profile specific information.								

7.3.5.2 Alarm class specification

7.3.5.2.1 Template

An Alarm object is described by the following template:

ASE: Alarm ASE
CLASS: Alarm
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

- 1 (m) Key Attribute: Implicit
- 2 (m) Attribute: Priority
- 3 (m) Attribute: Alarm Priority High
- 3.1 (m) Attribute: Sequence Number
- 4 (m) Attribute: Alarm Priority Low
- 4.1 (m) Attribute: Sequence Number

SERVICES:

- 1 (m) OpsService: Alarm Notification

7.3.5.2.2 Attributes

Implicit

The key attribute Implicit indicates that the Alarm object is implicitly addressed by the services.

Priority

This attribute defines whether the Alarm ASE uses one or two alarm queues. Note that this attribute is controlled over the alarm CR in the CRL class.

Allowed values: USER_PRIORITY - the alarm priority as given by the user is used and Alarm notifications are enqueued in the alarm high or alarm low queue accordingly,
 FIXED_PRIORITY - user priority is ignored and Alarm notifications are always enqueued in the alarm low queue accordingly only the alarm low queue is used.

An IO device as alarm sink shall always support USER_PRIORITY and FIXED_PRIORITY. The IO controller as alarm sink may choose whether it limits itself to one resource or not.

Alarm Priority High

This attribute consists of the following attributes:

Sequence Number

This attribute stores the sequence number of the last Alarm notification being send to the alarm sink. It is initialized with an arbitrary value and shall be incremented with every Alarm notification.

Whenever an Alarm is repeated over different ARs of an ARSet the same Sequence Number shall be used.

This attribute is not used in the context of a System redundancy notification.

Attribute type: Unsigned16

Allowed values: 0x0000 – 0x7FFF

Alarm Priority Low

This attribute consists of the following attributes:

Sequence Number

This attribute stores the sequence number of the last Alarm notification being send to the alarm sink. It is initialized with an arbitrary value and shall be incremented with every Alarm notification.

Whenever an Alarm is repeated over different ARs of an ARSet the same Sequence Number shall be used.

This attribute is not used in the context of a System redundancy notification.

Attribute type: Unsigned16

Allowed values: 0x0000 – 0x7FFF

7.3.5.2.3 Invocation of the Alarm object

Each AR has its own Alarm object.

7.3.5.3 Alarm service specification

7.3.5.3.1 Alarm Notification

This service is used to transfer an alarm notification from the IO device to the IO controller and optional vice versa. This service shall only be used in conjunction with the IO AR. Table 162 shows the parameters of the service.

Table 162 – Alarm Notification

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Alarm Priority	M	M(=)		
Alarm type	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Module Ident Number	M	M(=)		

Parameter name	req	ind	rsp	cnf
Submodule Ident Number	M	M(=)		
Alarm Specifier	M	M(=)		
Sequence Number	M	M(=)		
Channel Diagnosis	U	U(=)		
Manufacturer Specific Diagnosis	U	U(=)		
Submodule Diagnosis State	U	U(=)		
AR Diagnosis State	U	U(=)		
Alarm Payload	U	U(=)		
Maintenance Item	U	U(=)		
User Structure Identifier	M	M(=)		
Maintenance Status	M	M(=)		
Alarm Item	M	M(=)		
User Structure Identifier	M	M(=)		
Diagnosis Item	C	C(=)		
Channel Diagnosis	C	C(=)		
List of Channel Diagnosis Data	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
Channel Error type	M	M(=)		
Ext Channel Diagnosis	C	C(=)		
List of Ext Channel Diagnosis Data	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
Channel Error type	M	M(=)		
Ext Channel Error type	M	M(=)		
Ext Channel Add Value	M	M(=)		
Qualified Channel Diagnosis	C	C(=)		
List of Qualified Channel Diagnosis Data	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
Channel Error type	M	M(=)		
Ext Channel Error type	M	M(=)		
Ext Channel Add Value	M	M(=)		
Qualified Channel Qualifier	M	M(=)		
Manufacturer Specific Diagnosis	C	C(=)		
List of Manufacturer Specific Diagnosis Data	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
User Structure Identifier	M	M(=)		
List of Data	M	M(=)		
Data	M	M(=)		
Upload & Retrieval Item	C	C(=)		

Parameter name	req	ind	rsp	cnf
List of Records	M	M(=)		
Record Number	M	M(=)		
Record Length	M	M(=)		
iParameter Item	C	C(=)		
iPar Req Header	M	M(=)		
Max Segm Size	M	M(=)		
Transfer Index	M	M(=)		
Total iPar Size	M	M(=)		
Manufacturer Alarm Item	C	C(=)		
List of Data	M	M(=)		
Data	M	M(=)		
Profile Alarm Item	C	C(=)		
List of Data	M	M(=)		
Data	M	M(=)		
Result(+)			S	S
AREP			M	M
Alarm type			M	M
API			M	M
Slot Number			M	M
Subslot Number			M	M
Alarm Specifier			M	M
PNIO Status			M	M
Result(-)			S	S
AREP			M	M
PNIO Status			M	M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Alarm Priority

This parameter contains the priority of the alarm.

Allowed Values: ALARM_HIGH, ALARM_LOW

Alarm type

Attribute type: Unsigned16

Allowed values: see Table 161

API

This parameter defines the API the submodule issuing the alarm belongs to.

Attribute type: Unsigned32

Allowed values: 0 as default application process; 1 to 0xFFFFFFFF are reserved for the use in profile guidelines.

Slot Number

This parameter defines the slot number as a local identifier. This number is unique within a device. The numbering of slots within a device shall be in ascending order but may contain gaps in order to allow logical structuring. It shall contain the physical or logical slot which issues the alarm.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Subslot Number

This parameter defines the subslot number as a local identifier. This number is unique within a device. The numbering of subslots within a slot shall be in ascending order but may contain gaps in order to allow logical structuring. It shall contain the physical or logical subslot which issues the alarm.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF are allowed

Module Ident Number

This parameter contains the module identification.

Attribute type: Unsigned32

Allowed Values: 1 to 0xFFFFFFFF

NOTE 1 The parameter value is manufacturer specific.

Submodule Ident Number

This parameter contains the submodule identification.

Attribute type: Unsigned32

NOTE 2 The parameter value is manufacturer specific.

Alarm Specifier

This parameter consists of the following parameters.

Sequence Number

This parameter defines the sequence number of the alarm. By means of the Sequence Number the receiver detects duplications and reflects the value of the field in the Alarm.

Allowed values: 0 to 2 047

Channel Diagnosis

This parameter shall only be used for Alarm types attached to the Diagnosis ASE (see Table 161). For all other Alarm Types this parameter is not valid.

Table 163 shows the allowed values.

Table 163 – Channel Diagnosis

Value	Meaning
NO_CHANNEL_DIAG	Means that the Subslot Number does not include a Channel Diagnosis
CHANNEL_DIAG	Means that the Subslot Number includes at least one Channel Diagnosis

Manufacturer Specific Diagnosis

This parameter shall only be used for Alarm types attached to the Diagnosis ASE (see Table 161). For all other Alarm Types this parameter is not valid.

Table 164 shows the allowed values.

Table 164 – Manufacturer Specific Diagnosis

Value	Meaning
NO_MAN_DIAG	Means that the Subslot Number does not include a Manufacturer Specific Diagnosis
MAN_DIAG	Means that the Subslot Number includes at least one Manufacturer Specific Diagnosis

Submodule Diagnosis State

This parameter shall only be used for Alarm types attached to the Diagnosis ASE (see Table 161). For all other Alarm Types this parameter is not valid.

Table 165 shows the allowed values.

Table 165 – Submodule Diagnosis State

Value	Meaning
NO_DIAG	Error free Furthermore, it indicates that all reported diagnosis have been cleared. An individual "disappears" notification can be omitted. However, even in this case a channel diagnosis locally treated as status and/or a Manufacturer Specific Diagnosis treated as status may be present.
DIAG	At least one diagnosis exists at the submodule. It could be one or more channel diagnosis and/or one or more Manufacturer Specific Diagnosis.

AR Diagnosis State

This parameter shall only be used for Alarm types attached to the Diagnosis ASE (see Table 161). For all other Alarm Types this parameter is not valid.

The AR Diagnosis State shall be corresponding to the current state of the IO Data ASE parameter Problem Indicator Flag, which is related to all modules/submodules that belong to the AR.

Table 166 shows the allowed values.

Table 166 – AR Diagnosis State

Value	Meaning
NO_DIAG	Error free Furthermore, it indicates that all reported diagnosis related to the AR have been cleared. An individual "disappears" notification can be omitted. However, even in this case a channel diagnosis locally treated as status and/or a Manufacturer Specific Diagnosis treated as status may be present.
DIAG	At least one diagnosis exists at one submodule related to the AR. It could be one or more channel diagnosis and/or one or more Manufacturer Specific Diagnosis.

Alarm Payload

This optional parameter contains the alarm payload; structured according to the rules shown in Table 161. It consists of the following parameters:

Maintenance Item

This optional parameter contains the maintenance status. It shall only be present in conjunction with alarm notifications attached to the diagnosis ASE and if at least one of the parameters below has maintenance or qualified information; otherwise it shall be omitted.

It consists of the following parameters:

User Structure Identifier

This parameter identifies the format of the item.

Attribute type: Unsigned16

Allowed values: MAINTENANCE_USI

Maintenance Status

This parameter contains the maintenance status of the submodule. It consists of the following parameters:

Maintenance Required

This parameter contains the value MAINTENANCE_REQUIRED if at least one channel of the alarm source (API, Slot, Subslot) contains maintenance required information.

Allowed Values: MAINTENANCE_REQUIRED,
NO_MAINTENANCE_REQUIRED

Maintenance Demanded

This parameter contains the value MAINTENANCE_DEMANDED if at least one channel of the alarm source (API, Slot, Subslot) contains maintenance demanded information.

Allowed Values: MAINTENANCE_DEMANDED,
NO_MAINTENANCE_DEMANDED

Qualifier 2

This parameter contains the value QUALIFIER_2_SET if at least one channel of the alarm source (API, Slot, Subslot) contains qualified information. The semantic shall be defined by profiles.

Allowed Values: QUALIFIER_2_SET, QUALIFIER_2_NOT_SET

Qualifier 31

This parameter contains the value QUALIFIER_31_SET if at least one channel of the alarm source (API, Slot, Subslot) contains qualified information. The semantic shall be defined by profiles.

Allowed Values: QUALIFIER_31_SET, QUALIFIER_31_NOT_SET

Alarm Item

This optional parameter contain the accompanying information for the alarm. It is present, if further information is conveyed. The rules which item (i.e. diagnosis item, upload and retrieval item, iParameter Item, or common alarm item) is conveyed are described in Table 161.

This parameter is composed of the following parameters.

User Structure Identifier

This parameter identifies the format of the item.

Attribute type: Unsigned16

Allowed values: shown in Table 167

Table 167 – User Structure Identifier

Value	Meaning / Usage
CHANNEL_DIAGNOSIS_USI	Shall only be used in conjunction with Channel Diagnosis in Alarm Notification and Diagnosis Data.
EXT_CHANNEL_DIAGNOSIS_USI	Shall only be used in conjunction with Extended Channel Diagnosis in Alarm Notification and Diagnosis Data.
QUALIFIED_CHANNEL_DIAGNOSIS_USI	Shall only be used in conjunction with Qualified Channel Diagnosis in Alarm Notification and Diagnosis Data.
MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF	In conjunction with alarm type Diagnosis Manufacturer Specific Diagnosis in Alarm Notification and Diagnosis Data. In conjunction with other alarm types the usage is manufacturer specific.
PROFILE_SPECIFIC_USI_0x9000 up to PROFILE_SPECIFIC_USI_0x9FFF	The meaning is defined in profile specifications.
UPLOAD_AND RETRIEVAL_USI	Shall be used in conjunction with Upload and Retrieval notification.
IPARAMETER_USI	Shall be used in conjunction with iParameter notification.

Diagnosis Item

This optional parameter is present only, if the User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, QUALIFIED_CHANNEL_DIAGNOSIS_USI, MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF.

A list element consists of the following parameters:

Channel Diagnosis

This optional parameter is present if the User Structure Identifier is set to CHANNEL_DIAGNOSIS_USI. It consists of the following parameters:

List of Channel Diagnosis Data

This parameter contains a list of channel diagnosis data. A list element consists of the following parameters:

Channel Number

This parameter defines the channel number as a local identifier. This number is unique within a subslot. The numbering of channels within a subslot shall be in ascending order but may contain gaps in order to allow logical structuring. The list shall contain all physical or logical channels which have diagnosis.

The special value 0x8000 identifies the whole submodule instead of a special channel.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF user specific channel number, 0x8000 submodule

Channel Properties

This parameter consists of the following parameters:

Type

This parameter has the same definition as specified in the Diagnosis ASE.

Accumulative

This parameter has the same definition as specified in the Diagnosis ASE.

Maintenance

This parameter has the same definition as specified in the Diagnosis ASE.

Specifier

This parameter has an extended definition compared to the one specified in the Diagnosis ASE.

This parameter indicates that diagnosis/maintenance appears or disappears. The semantic is shown in Table 168.

Allowed values: APPEARS, DISAPPEARS, DISAPPEARS_BUT_OTHER_REMAIN, ALL_DISAPPEARS

Table 168 – Semantics of Specifier

Maintenance	Specifier	Meaning
DIAGNOSIS	APPEARS	Diagnosis appears The diagnosis ASE contains this and possible other entries with ChannelProperties.Maintenance = DIAGNOSIS for this channel.
DIAGNOSIS	DISAPPEARS	Diagnosis disappears The diagnosis ASE contains no further entry with ChannelProperties.Maintenance = DIAGNOSIS for this channel.
DIAGNOSIS	DISAPPEARS_BUT_OTHER_REMAIN	Diagnosis disappears The diagnosis ASE still contains other entries with ChannelProperties.Maintenance = DIAGNOSIS for this channel.
DIAGNOSIS	ALL_DISAPPEARS	Diagnosis disappears The diagnosis ASE contains no longer any entries for this channel, all subsequent ^a Diagnosis, Maintenance (required and demanded) and Qualified disappears.
MAINTENANCE_REQUIRED	APPEARS	Maintenance Required appears The diagnosis ASE contains this and possible other entries with ChannelProperties.Maintenance = MAINTENANCE_REQUIRED for this channel.
MAINTENANCE_REQUIRED	DISAPPEARS	Maintenance Required disappears The diagnosis ASE contains no further entry with ChannelProperties.Maintenance = MAINTENANCE_REQUIRED for this channel.
MAINTENANCE_REQUIRED	DISAPPEARS_BUT_OTHER_REMAIN	Maintenance Required disappears The diagnosis ASE still contains other entries with ChannelProperties.Maintenance = MAINTENANCE_REQUIRED for this channel.

Maintenance	Specifier	Meaning
MAINTENANCE_DEMANDED	APPEARS	Maintenance Demanded appears The diagnosis ASE contains this and possible other entries with ChannelProperties.Maintenance = MAINTENANCE_DEMANDED for this channel.
MAINTENANCE_DEMANDED	DISAPPEARS	Maintenance Demanded disappears The diagnosis ASE contains no further entry with ChannelProperties.Maintenance = MAINTENANCE_DEMANDED for this channel.
MAINTENANCE_DEMANDED	DISAPPEARS_BUT_OTHER_REMAIN	Maintenance Demanded disappears The diagnosis ASE still contains other entries with ChannelProperties.Maintenance = MAINTENANCE_DEMANDED for this channel.
QUALIFIED_DIAGNOSIS	APPEARS	Qualified Diagnosis appears The diagnosis ASE contains this and possible other entries with ChannelProperties.Maintenance = QUALIFIED_DIAGNOSIS for this channel.
QUALIFIED_DIAGNOSIS	DISAPPEARS	Qualified Diagnosis disappears The diagnosis ASE contains no further entry with ChannelProperties.Maintenance = QUALIFIED_DIAGNOSIS for this channel.
QUALIFIED_DIAGNOSIS	DISAPPEARS_BUT_OTHER_REMAIN	Qualified Diagnosis disappears The diagnosis ASE still contains other entries with ChannelProperties.Maintenance = QUALIFIED_DIAGNOSIS for this channel.
^a Subsequent means, that for this particular ChannelErrorType all ExtChannelErrorTypes (and the ChannelErrorType itself) with diagnosis, maintenance required, maintenance demanded, and qualified changes to OK.		

Direction

This parameter has the same definition as specified in the Diagnosis ASE.

Channel Error type

This parameter has the same definition as specified in the Diagnosis ASE.

Ext Channel Diagnosis

This optional parameter is present if the User Structure Identifier is set to EXT_CHANNEL_DIAGNOSIS_USI. It consists of the following parameters:

List of Ext Channel Diagnostic Data

This parameter contains a list of extended channel diagnosis data. A list element consists of the following parameters:

Channel Number

This parameter has the same definition as specified in the Diagnosis ASE.

Channel Properties

This parameter has the same definition as specified in the Channel Diagnosis above.

Channel Error type

This parameter has the same definition as specified in the Diagnosis ASE.

Ext Channel Error type

This parameter has the same definition as specified in the Diagnosis ASE.

Ext Channel Add Value

This parameter has the same definition as specified in the Diagnosis ASE.

Qualified Channel Diagnosis

This optional parameter is present if the User Structure Identifier is set to QUALIFIED_CHANNEL_DIAGNOSIS_USI. It consists of the following parameters:

List of Qualified Channel Diagnostic Data

This parameter contains a list of extended channel diagnosis data. A list element consists of the following parameters:

Channel Number

This parameter has the same definition as specified in the Diagnosis ASE.

Channel Properties

This parameter has the same definition as specified in the Channel Diagnosis above.

Channel Error type

This parameter has the same definition as specified in the Diagnosis ASE.

Ext Channel Error type

This parameter has the same definition as specified in the Diagnosis ASE.

Ext Channel Add Value

This parameter has the same definition as specified in the Diagnosis ASE.

Qualified Channel Qualifier

This parameter has the same definition as specified in the Diagnosis ASE.

Manufacturer Specific Diagnosis

This optional parameter is present if the User Structure Identifier is set to a value in the range from MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF. It consists of the following parameters:

Manufacturer specific structures shall only be used if Channel Diagnosis, or Extended Channel Diagnosis or Qualified Extended Channel Diagnosis does not cover the diagnosis information.

List of Manufacturer Specific Diagnostic Data

This parameter contains a list of manufacturer specific channel diagnosis data. A list element consists of the following parameters:

API

This parameter defines the API the submodule issuing the alarm belongs to.

Attribute type: Unsigned32

Allowed values: 0 as default application process; 1 to 0xFFFFFFFF are reserved for the use in profile guidelines.

Slot Number

This parameter defines the slot number as a local identifier. This number is unique within a device. The numbering of slots within a device shall be in ascending order but may contain gaps in order to allow logical structuring. It shall contain the physical or logical slot which issues the alarm.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Subslot Number

This parameter defines the subslot number as a local identifier. This number is unique within a device. The numbering of subslots within a slot shall be in ascending order but may contain gaps in order to allow logical structuring. It shall contain the physical or logical subslot which issues the alarm.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF are allowed

Channel Number

This parameter has the same definition as specified in the Diagnosis ASE.

Channel Properties

This parameter has the same definition as specified in the Channel Diagnosis above.

User Structure Identifier

This parameter indicates the structure of the Manufacturer Specific Diagnosis Data parameter.

Attribute type: Unsigned16

Allowed values: MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF

List of Data

This parameter contains a list of manufacturer specific diagnosis data. A list element consists of the following parameters.

Data

This parameter contains manufacturer specific diagnosis data.

Attribute type: One of the following data types shall be used:
Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference,
Floating Point, Network Time, VisibleString, Network Time Difference,
OctetString

Upload and Retrieval Item

This optional parameter is present only, if the Alarm Type is set to "Upload and Retrieval Notification" and the User Structure Identifier is set to UPLOAD_AND_RETRIEVAL_USI.

With this parameter a submodule indicates the existence of Record Data objects which shall be stored at the IO controller. The IO controller shall upload and permanently store the Record Data objects. At every following connect sequence

and for every following plug and released sequence the IO controller shall convey this additional Record Data objects to the source of the stored Record Data objects. Record Data objects which are part of the GSD description and conveyed from the engineering system to the IO controller shall not be part of the List of Records.

It consists of the following parameters:

List of Records

This parameter contains a list of Record Data objects. A list element consists of the following parameters:

Record Number

This parameter contains the Index of a Record Data object as shown in the Record Data ASE.

Allowed Values: 0 to 0xFFFF

Record Length

This parameter contains the Length of a Record Data object as shown in the Record Data ASE.

Allowed Values: 2^0 to $2^{32}-256$

iParameter Item

This optional parameter is present only, if the Alarm Type is set to "Upload and Retrieval Notification" and the User Structure Identifier is set to IPARAMETER_USI.

With this parameter a submodule indicates to upload iParameters or requests to download previously stored iParameters.

The difference between this iParameter Item and the Upload and Retrieval Item defined above is that the iParameter Item uses one record data element, allows segmented transfer of the record data element and transfers the record data on request only.

If the submodule requests an upload, the IO controller shall upload and permanently store the record data object indicated in the alarm.

If the submodule requests a download, the previously stored record data object (result of a previous upload) is transferred to the submodule.

Record Data objects which are part of the GSD description and conveyed from the engineering system to the IO controller shall not be part of this iParameter record data object.

It consists of the following parameters:

iPar Req Header

This parameter contains the request header used for the up-/download of the iParameters. It consists of the following parameters:

SR_Version

This parameter indicates the version of the iParameter protocol.

Attribute type: Unsigned8

Allowed values: 0x01

N_Count

This parameter indicates the notification count of the iParameter protocol.

Attribute type: Unsigned8

Allowed values: 0x01

SR_Type

This parameter indicates the subtype of the iParameter notification.

Attribute type: Unsigned8

Allowed values: 0x01 for requesting an upload, 0x03 for requesting a download

Max Segm Size

This parameter contains the maximum segment size to use for an individual record data read. It thus allows fragmentation of huge iParameters.

Attribute type: Unsigned32

Allowed Values: 2^0 to $2^{22}-1$

Transfer Index

This parameter contains the record data index to use for the up-/download of the iParameters.

Attribute type: Unsigned32

Allowed values: 0 – 0x7FFF

Total iPar Size

This parameter contains the total size of the iParameters. The parameter shall be set to the size of the iParameters when requesting an upload. On requesting a download the value “0” is allowed; indicating to download the whole parameter set as stored in the IO controller.

Attribute type: Unsigned32

Allowed Values: 2^0 to $2^{32}-256$

Manufacturer Alarm Item

This optional parameter is present only, if the Alarm Type is set to “Manufacturer Specific”. In this case the User Structure Identifier shall be set to MANUFACTURER_SPECIFIC_USI_0x0000 up to MANUFACTURER_SPECIFIC_USI_0x7FFF.

List of Data

This parameter contains a list of manufacturer specific alarm data.

The length of the user data shall not exceed 1 408 octets.

A list element is composed of the following parameters:

Data

This parameter contains user specific alarm data.

Attribute type: One of the following data types shall be used:

Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference, Floating Point, Network Time, VisibleString, Network Time Difference, OctetString

Profile Alarm Item

This optional parameter is present only, if the Alarm Type is set to “Profile Specific”. In this case the User Structure Identifier shall be set to PROFILE_SPECIFIC_USI_0x9000 up to PROFILE_SPECIFIC_USI_0x9FFF.

List of Data

This parameter contains a list of profile specific alarm data.

The length of the user data shall not exceed 1 408 octets.

A list element is composed of the following parameters:

Data

This parameter contains user specific alarm data.

Attribute type: One of the following data types shall be used:

Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference, Floating Point, Network Time, VisibleString, Network Time Difference, OctetString

Result(+)

This parameter indicates that the service request succeeded.

Alarm type

This parameter reflects the value from the request.

API

This parameter reflects the value from the request.

Slot Number

This parameter reflects the value from the request.

Subslot Number

This parameter reflects the value from the request.

Alarm Specifier

This parameter reflects the value from the request.

PNIO Status

This parameter contains the error status.

Allowed Values: NO_ERROR, ALARM_TYPE_NOT_SUPPORTED, WRONG_SUBMODULE_STATE, BACKUP_ALARM_NOT_EXECUTED

Attribute type: Unsigned32

Result(-)

This parameter indicates that the service request failed.

PNIO Status

This parameter contains the reason for the failure.

7.3.5.4 Behavior of Alarm objects**7.3.5.4.1 General behavior of the alarm sink**

The application behavior based on flow control of alarm transmission, IOPS, IOCS and Problem Indicator of the IO Data ASE. These mechanisms make it convenient for the application to process the alarm.

NOTE The application process may not handle the alarm immediately by interrupting other processes. It may also poll for queued alarms to handle these in an appropriate time window.

Each application process shall maintain a receiving queue for each Alarm type. If used the queue size is determined by the flow control mechanisms. With the processing of the alarm the acknowledgment shall be sent to the source to enable the sending of further alarms of this type.

Alarm queues shall be cleared in case of AR termination and pull/plug alarms.

7.3.5.4.2 General behavior of the alarm source

Figure 32 shows the model at the alarm source. The source is divided into the IO device (IOD), the application process and the alarm source.

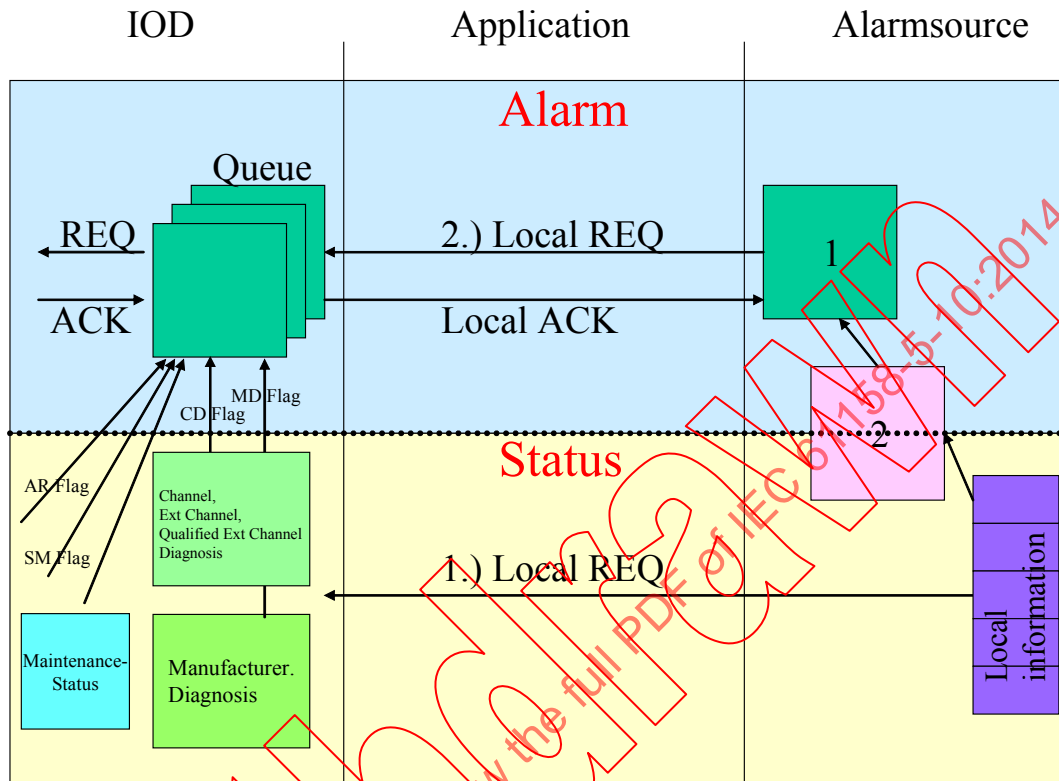


Figure 32 – Example of a resource model at the alarm source

The model shows how to handle event overflow at the source. Event overflow may cause by flow control because the sink determines how fast alarms are handled. A mixture of queue and buffer model solves that problem for the Alarm ASE.

The alarm source maintains local information of the state. If, for example, the status changes from GOOD to BAD it is notified. It maintains at least two buffers for alarms. The first buffer is used to transport the state transition. The second buffer contains the current state. A new state transition will not be reported as long as the acknowledgement is not received in case of only two buffers. But the current state may be read with diagnosis information at any time.

The IO device maintains two alarm queues, one for priority ALARM_HIGH and one for priority ALARM_LOW. The IO controller may request over the alarm CR that only the alarm queue for priority ALARM_LOW is used by the IO device.

A new alarm of the same priority is blocked as long as the acknowledgement is outstanding. Blocking the alarms with the same priority by waiting for the Alarm_ACK from the IO controller queues up further alarms in an alarm queue at the IO device. This queue may contain entries which are obsolete due to other (later) entries (e.g. diagnosis “short circuit” coming and going when having a loose cable). In such a case the IO device may delete entries as an optimization if this deletion doesn't affect the user.

Furthermore, alarm queues shall be cleared in case of AR termination and pull/plug alarms.

7.3.5.4.3 Behavior for diagnosis changes

Diagnosis alarms are the means to report and acknowledge diagnosis changes. It is strongly recommended to use Channel Diagnoses or Extended Channel Diagnosis because of the well-defined structure.

The recommended sequence for signaling a diagnosis is:

- First, update the diagnosis ASE
- Second, issue an alarm.

Diagnosis alarms include information of the alarm source and the direction of changes. The attribute Alarm Specifier contains information about the direction e.g. appears or disappears.

All channel diagnosis are gone is reported by an alarm with the parameter Alarm Specifier Submodule Diagnosis State "NO_ERROR". It may save a lot of individual messages.

7.3.6 Context ASE

7.3.6.1 Overview

The Context ASE provides a set of services to establish application relationships, parameterize and configure the AP of an IO device itself and its slots/subslots containing IO data.

The Application Process of IO devices may need parameter (e.g. the range of an analogue signal) from the assigned IO controller to provide the data from the Input Data object and/or for the Output Data object according to the requirements of the AP of the IO controller. These IO devices provide in the Context ASE an IO AR Parameter object or, for more flexibility, several IO AR Structured Parameter objects. The values for the attributes of these objects will be provided by the IO controller and assigned after validation from the AP of the IO device.

The functionality of an IO device application implies the necessary parameterization objects (specific Record Data objects) which have to be conveyed at the start-up phase of the system from the IO controller to the assigned IO devices via the IO AR. The application of the IO device has to check the parameterization data sent by the IO controller.

NOTE In comparison to the IO Data ASE, where all the actual available slots and subslots with their attributes are stored, the Context ASE contains only the configured or used slots and subslots after a successful connection establishment.

7.3.6.2 Context class specification

7.3.6.2.1 Template

This class specifies the object for the parameter data that are related to the IO devices itself and/or to the slots/subslots of the IO device.

The IO Device context object is described by the following template:

ASE:	Context ASE
CLASS:	Context
CLASS ID:	not used
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: List of APs
2.1	(m) Attribute: API
2.2	(m) Attribute: Real Identification

2.2.1	(m) Attribute:	List of Slots
2.2.1.1	(m) Attribute:	Slot Number
2.2.1.2	(m) Attribute:	Module Ident Number
2.2.1.3	(m) Attribute:	Module Properties
2.2.1.4	(m) Attribute:	List of Subslots
2.2.1.4.1	(m) Attribute:	Subslot Number
2.2.1.4.2	(m) Attribute:	Submodule Ident Number
2.2.1.4.3	(m) Attribute:	Submodule Properties
2.2.1.4.3.1	(m) Attribute:	Type
2.2.1.4.3.2	(m) Attribute:	Shared Input
2.2.1.4.3.3	(m) Attribute:	Reduce Input Submodule Data Length
2.2.1.4.3.4	(m) Attribute:	Reduce Output Submodule Data Length
2.2.1.4.3.5	(m) Attribute:	Discard IOXS
2.2.1.4.4	(m) Attribute:	List of Data Descriptions
2.2.1.4.4.1	(m) Attribute:	Data Direction
2.2.1.4.4.2	(m) Attribute:	Submodule Data Length
2.2.1.4.4.3	(m) Attribute:	Length IOPS
2.2.1.4.4.4	(m) Attribute:	Length IOCS
2.3	(m) Attribute:	Expected Identification
2.3.1	(m) Attribute:	List of ARs
2.3.1.1	(m) Attribute:	AREP
2.3.1.2	(o) Attribute:	Sync State
2.3.1.3	(m) Attribute:	List of CRs
2.3.1.3.1	(m) Attribute:	GREP
2.3.1.3.2	(m) Attribute:	Multicast Provider State
2.3.1.4	(m) Attribute:	List of APIs
2.3.1.4.1	(m) Attribute:	API
2.3.1.4.2	(m) Attribute:	List of Slots
2.3.1.4.2.1	(m) Attribute:	Slot Number
2.3.1.4.2.2	(m) Attribute:	Module Ident Number
2.3.1.4.2.3	(m) Attribute:	Module Properties
2.3.1.4.2.4	(m) Attribute:	Module State
2.3.1.4.2.5	(m) Attribute:	List of Subslots
2.3.1.4.2.5.1	(m) Attribute:	Subslot Number
2.3.1.4.2.5.2	(m) Attribute:	Submodule Ident Number
2.3.1.4.2.5.3	(m) Attribute:	Submodule Properties
2.3.1.4.2.5.3.1	(m) Attribute:	Type
2.3.1.4.2.5.3.2	(m) Attribute:	Shared Input
2.3.1.4.2.5.4	(m) Attribute:	Submodule State
2.3.1.4.2.5.4.1	(m) Attribute:	Format Indicator
2.3.1.4.2.5.4.2	(c) Constraint:	Format Indicator = V_1_0
2.3.1.4.2.5.4.2.1	(m) Attribute:	Detail
2.3.1.4.2.5.4.3	(c) Constraint:	Format Indicator <> V_1_0
2.3.1.4.2.5.4.3.1	(m) Attribute:	Add Info
2.3.1.4.2.5.4.3.2	(m) Attribute:	Qualifier Info
2.3.1.4.2.5.4.3.3	(m) Attribute:	Maintenance Required
2.3.1.4.2.5.4.3.4	(m) Attribute:	Maintenance Demanded
2.3.1.4.2.5.4.3.5	(m) Attribute:	Diag Info
2.3.1.4.2.5.4.3.6	(m) Attribute:	AR Info
2.3.1.4.2.5.4.3.7	(m) Attribute:	Ident Info
3	(m) Attribute:	I&M Data
3.1	(m) Attribute:	List of Slots
3.1.1	(m) Attribute:	Slot Number

3.1.2	(m) Attribute:	List of Subslots
3.1.2.1	(m) Attribute:	Subslot Number
3.1.2.2	(m) Attribute:	I&M0 Data
3.1.2.2.1	(m) Attribute:	Vendor ID
3.1.2.2.2	(m) Attribute:	Order ID
3.1.2.2.3	(m) Attribute:	Serial Number
3.1.2.2.4	(m) Attribute:	Hardware Revision
3.1.2.2.5	(m) Attribute:	Software Revision
3.1.2.2.5.1	(m) Attribute:	Prefix
3.1.2.2.5.2	(m) Attribute:	Functional Enhancement
3.1.2.2.5.3	(m) Attribute:	Bug Fix
3.1.2.2.5.4	(m) Attribute:	Internal Change
3.1.2.2.6	(m) Attribute:	Revision Counter
3.1.2.2.7	(m) Attribute:	Profile ID
3.1.2.2.8	(m) Attribute:	Profile Specific Type
3.1.2.2.9	(m) Attribute:	I&M Version
3.1.2.2.10	(m) Attribute:	I&M Supported
3.1.2.3	(o) Attribute:	I&M1 Data
3.1.2.3.1	(m) Attribute:	Function
3.1.2.3.2	(m) Attribute:	Location
3.1.2.4	(o) Attribute:	I&M2 Data
3.1.2.4.1	(m) Attribute:	Installation Date
3.1.2.5	(o) Attribute:	I&M3 Data
3.1.2.5.1	(m) Attribute:	Description
3.1.2.6	(o) Attribute:	I&M4 Data
3.1.2.6.1	(m) Attribute:	Signature
4	(m) Attribute:	I&M0 Filter Data
4.1	(m) Attribute:	I&M0 Carrier Data
4.1.1	(m) Attribute:	List of APs
4.1.1.1	(m) Attribute:	API
4.1.1.2	(m) Attribute:	List of Slots
4.1.1.2.1	(m) Attribute:	Slot Number
4.1.1.2.2	(m) Attribute:	Module Ident Number
4.1.1.2.3	(m) Attribute:	List of Subslots
4.1.1.2.3.1	(m) Attribute:	Subslot Number
4.1.1.2.3.2	(m) Attribute:	Submodule Ident Number
4.2	(o) Attribute:	Module Representative Data
4.2.1	(m) Attribute:	List of APs
4.2.1.1	(m) Attribute:	API
4.2.1.2	(m) Attribute:	List of Slots
4.2.1.2.1	(m) Attribute:	Slot Number
4.2.1.2.2	(m) Attribute:	Module Ident Number
4.2.1.2.3	(m) Attribute:	List of Subslots
4.2.1.2.3.1	(m) Attribute:	Subslot Number
4.2.1.2.3.2	(m) Attribute:	Submodule Ident Number
4.3	(m) Attribute:	Device Representative Data
4.3.1	(m) Attribute:	List of APs
4.3.1.1	(m) Attribute:	API
4.3.1.2	(m) Attribute:	List of Slots
4.3.1.2.1	(m) Attribute:	Slot Number
4.3.1.2.2	(m) Attribute:	Module Ident Number
4.3.1.2.3	(m) Attribute:	List of Subslots
4.3.1.2.3.1	(m) Attribute:	Subslot Number

4.3.1.2.3.2	(m) Attribute:	Submodule Ident Number
5	(o) Attribute:	Autoconfiguration Data
5.1	(m) Attribute:	Communication Data
5.1.1	(m) Attribute:	Min Device Interval
5.1.2	(m) Attribute:	Communication Properties
5.1.2.1	(m) Attribute:	DFP
5.1.2.2	(m) Attribute:	RTC3
5.1.3	(m) Attribute:	Send Clock Factors
5.2	(m) Attribute:	Configuration Data
5.2.1	(m) Attribute:	List of Offered Submodules
5.2.1.1	(m) Attribute:	API
5.2.1.2	(m) Attribute:	Slot Number
5.2.1.3	(m) Attribute:	Subslot Number
5.2.1.4	(m) Attribute:	Module Ident Number
5.2.1.5	(m) Attribute:	Submodule Ident Number
5.2.1.6	(m) Attribute:	Length Input Data
5.2.1.7	(m) Attribute:	Length Output Data
5.3	(o) Attribute:	Isochronous Data
5.3.1	(m) Attribute:	List of Offered Submodules
5.3.1.1	(m) Attribute:	API
5.3.1.2	(m) Attribute:	Slot Number
5.3.1.3	(m) Attribute:	Subslot Number
5.3.1.4	(m) Attribute:	Time IO Base
5.3.1.5	(m) Attribute:	Time IO Input
5.3.1.6	(m) Attribute:	Time IO Output
6	(o) Attribute:	Fast Startup Data
6.1	(m) Attribute:	List of ARs
6.1.1	(m) Attribute:	AREP
6.1.2	(m) Attribute:	Expected Fast Startup Data
6.1.2.1	(o) Attribute:	FS Parameter Block
6.1.2.1.1	(m) Attribute:	FS Parameter Mode
6.1.2.1.2	(m) Attribute:	FS Parameter UUID
6.1.2.2	(o) Attribute:	Manufacturer Specific Fast Startup Data
7	(o) Attribute:	GSD Data
SERVICES:		
1	(m) OpsService:	Connect
2	(m) OpsService:	Connect Device Access
3	(m) OpsService:	Release
4	(o) OpsService:	Prm Begin
5	(m) OpsService:	Prm End
6	(m) OpsService:	Application Ready
7	(m) OpsService:	Read Expected Identification Data for one subslot
8	(m) OpsService:	Read Real Identification Data for one subslot
9	(m) OpsService:	Read Expected Identification Data for one slot
10	(m) OpsService:	Read Real Identification Data for one slot
11	(m) OpsService:	Read Expected Identification Data for one AR
12	(m) OpsService:	Read Real Identification Data for one AR
13	(m) OpsService:	Read Real Identification Data for one API
14	(m) OpsService:	Read Module Diff Block
15	(m) OpsService:	Abort
16	(m) OpsService:	Local AR Abort
17	(m) OpsService:	Read AP Data
18	(m) OpsService:	Read I&M0 Filter Data

- 19 (m) OpsService: Read I&M0 Data
- 20 (o) OpsService: Write I&M1 Data
- 21 (o) OpsService: Read I&M1 Data
- 22 (o) OpsService: Write I&M2 Data
- 23 (o) OpsService: Read I&M2 Data
- 24 (o) OpsService: Write I&M3 Data
- 25 (o) OpsService: Read I&M3 Data
- 26 (o) OpsService: Write I&M4 Data
- 27 (o) OpsService: Read I&M4 Data
- 28 (o) OpsService: Read Autoconfiguration Data
- 29 (o) OpsService: Write Expected Fast Startup Data
- 30 (o) OpsService: Read Expected Fast Startup Data
- 31 (o) OpsService: Read GSD Data

7.3.6.2.2 Attributes

Implicit

The key attribute Implicit indicates that the User Parameter object is implicitly addressed by the services.

List of APs

One API is composed of the following list elements:

API

This attribute defines the number of the application process.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

Real Identification

This attribute consists of the following attributes:

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This attribute defines to which module the Real Identification Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 7FFF

Module Ident Number

This attribute defines the identification of the slot.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFFFFFF

NOTE 1 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Module Properties

This attribute is reserved for future use.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the Real Identification Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF are allowed

Submodule Ident Number

This attribute defines the identification of the subslot.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

NOTE 2 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Submodule Properties

This attribute consists of the following attributes.

Type

This attribute defines the type of the subslot.

Attribute type: Unsigned16

Allowed values: NO_IO, INPUT, OUTPUT, IO

Shared Input

This attribute defines the properties of the subslot. The value SHARED_INPUT shall only be used if the attribute Submodule type contains the value INPUT or IO.

Attribute type: Unsigned8

Allowed values: SHARED_INPUT, NOT_SHARED_INPUT

Reduce Input Submodule Data Length

This attribute defines whether to use the expected input submodule data length for the input CR or to reduce the data length to zero.

Attribute type: Unsigned8

Allowed values: EXPECTED, ZERO

Reduce Output Submodule Data Length

This attribute defines whether to use the expected output submodule data length for the output CR or to reduce the data length to zero.

Attribute type: Unsigned8

Allowed values: EXPECTED, ZERO

Discard IOXS

This attribute defines whether to transfer all expected IOXS for this submodule in the corresponding CRs or not.

Attribute type: Unsigned8

Allowed values: EXPECTED, ZERO

List of Data Descriptions

This attribute list consists of the following attributes.

Data Direction

The attribute defines the direction input or output.

Attribute type: Unsigned16

Allowed values: DIRECTION_INPUT, DIRECTION_OUTPUT

Submodule Data Length

The attribute Submodule Data Length defines the number of octets of the Data Element without counting the IOPS.

Attribute type: Unsigned16

Allowed values: 0 to 1 439

Length IOPS

The attribute Length IOPS defines the number of octets of the provider status.

Attribute type: Unsigned8

Allowed values: 1, (2 to 255 reserved)

Length IOCS

The attribute Length IOCS defines the number of octets of the consumer status.

Attribute type: Unsigned8

Allowed values: 1, (2 to 255 reserved)

Expected Identification

This attribute consists of the following attributes:

List of ARs

One AR is composed of the following list elements:

AREP

This parameter is the local identifier for the desired AR.

SYNC State

This attribute is the local synchronization status for the desired AR. It shall only be present if synchronization is used. The behavior shall be according to 7.3.7.

Attribute type: Unsigned16

Allowed values: SYNCHRONIZED, SYNC_NOT_AVAILABLE

List of CRs

One CR is composed of the following list elements:

CREP

This parameter is the local identifier for the desired multicast consumer CR.

Multicast Provider State

This parameter is the local status for the multicast consumer CR. The behavior shall be according to 7.3.7.

Attribute type: Unsigned16

Allowed values: UP_AND_RUNNING,
MULTICAST_PROVIDER_NOT_AVAILABLE

List of APIs

One API is composed of the following list elements:

API

This attribute defines the number of the application process.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

List of Slots

One Slot is composed of the following list elements:

Slot Number

This attribute defines to which module the IO Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Module Ident Number

This attribute defines the identification of the slot.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFFFFFF

NOTE 3 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Module Properties

This attribute is reserved for further use.

Module State

This attribute defines the state of the module and shall be used for the connect service and the application ready service.

Attribute type: Unsigned16

Allowed values: Shall be set according to Table 169.

Table 169 – Module State

Value	Use
NO_MODULE	The slot exists and the module is not plugged.
WRONG_MODULE	The module ident number is wrong. This coding should be avoided; use "Substitute" for the module and "Wrong (WR)" for each submodule
PROPER_MODULE	The slot exists and the module is okay.
SUBSTITUTE	The slot exists and the module is not the same as requested - but compatible. The IO device was able to adapt by its own decision. This coding should be used in conjunction with "Wrong (WR)" for each submodule

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the IO Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

Submodule Ident Number

This attribute defines the identification of the subslot.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

NOTE 4 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Submodule Properties

This attribute consists of the following attributes.

Type

This attribute defines the type of the subslot.

Attribute type: Unsigned16

Allowed values: NO_IO, INPUT, OUTPUT, IO

Table 170 shows the dependencies that shall be applied in conjunction with selected CR attribute CR type.

Table 170 – Usage with respect to CR type

Submodule type	Usage with CR type
NO_IO	INPUT CR, OUTPUT CR
INPUT	INPUT CR, OUTPUT CR, MULTICAST PROVIDER CR
OUTPUT	INPUT CR, OUTPUT CR, MULTICAST CONSUMER CR
IO	INPUT CR, OUTPUT CR, MULTICAST PROVIDER CR (Input part only), MULTICAST CONSUMER CR (Output part only)

Shared Input

This attribute defines the properties of the subslot. The value SHARED_INPUT shall only be used if the attribute Submodule type contains the value INPUT or IO.

Attribute type: Unsigned8

Allowed values: SHARED_INPUT, NOT_SHARED_INPUT

Reduce Input Submodule Data Length

This attribute defines whether to use the expected input submodule data length for the input CR or to reduce the data length to zero.

Attribute type: Unsigned8

Allowed values: EXPECTED, ZERO

Reduce Output Submodule Data Length

This attribute defines whether to use the expected output submodule data length for the output CR or to reduce the data length to zero.

Attribute type: Unsigned8

Allowed values: EXPECTED, ZERO

Discard IOXS

This attribute defines whether to transfer all expected IOXS for this submodule in the corresponding CRs or not.

Attribute type: Unsigned8

Allowed values: EXPECTED, ZERO

Submodule State

This attribute consists of the following attributes:

Format Indicator

This attribute indicates the format of the Submodule State.

Allowed Values: 0 (meaning = V_1_0), 1 (meaning > V_1_0)

Detail

This attribute contains the value for the submodule state (Format Indicator = V_1_0).

Shall be set according to Table 171.

Table 171 – Detail

Value	Meaning
NO_SUBMODULE	The subslot exists and is empty.
WRONG_SUBMODULE	The subslot exists and an incompatible submodule is plugged.
LOCKED_BY_IO_CONTROLLER	The subslot exists and is locked by another IO controller.
APPLICATION_READY_PENDING	This detail is issued statically or dynamically. Static means that the submodule exists and a parameter fault was detected or the application of the submodule is not operating. Dynamic means that the submodule exists but the execution of the parameterization is not ready yet. In this case the submodule will issue a Return of Submodule alarm, when ready.
SUBSTITUTE	The subslot exists and the submodule is not the same as requested – but compatible and the IO device was able to adapt In this case the IO device has to adapt e.g. to new input or output length
GOOD	The subslot exists and the expected submodule is plugged.

NOTE 5 The field Submodule State is only responded if there is a difference between expected and real configuration data. Submodule State “GOOD” is not defined because for such submodules no status is reported within the response.

Add Info

This attribute contains additional information for the application relation (Format Indicator > V_1_0).

For an application relation with AR Type IOCAR or IOCAR_SR the value is always set to NO_ADD_INFO.

For an application relation with AR Type IOSAR the value TAKEOVER_NOT_ALLOWED indicates that the submodule was configured with AR Properties “Supervisor Takeover allowed” set to “false” and thus the IO supervisor got no ownership for the submodule.

Allowed Values: NO_ADD_INFO, TAKEOVER_NOT_ALLOWED

Qualified Info

This attribute indicates whether qualified diagnosis information is present in at least one channel of the submodule (Format Indicator > V_1_0).

Qualifier Info is derived from the Diagnosis ASE; see 7.3.4.4.5 for details.

Allowed Values: NO_QUALIFIED_INFO, QUALIFIED_INFO

NOTE 6 If Shared Input contains the value SHARED_INPUT, this field contains always the value NO_QUALIFIED_INFO (see 7.4.6.2.2.1 for details).

Maintenance Required

This attribute indicates whether maintenance required is present in at least one channel of the submodule (Format Indicator > V_1_0).

Maintenance Required is derived from the Diagnosis ASE; see 7.3.4.4.3 for details.

Allowed Values: NO_MAINTENANCE_REQUIRED,
MAINTENANCE_REQUIRED

NOTE 7 If Shared Input contains the value SHARED_INPUT, this field contains always the value NO_MAINTENANCE_REQUIRED (see 7.4.6.2.2.1 for details).

Maintenance Demanded

This attribute indicates whether maintenance demanded is present in at least one channel of the submodule (Format Indicator > V_1_0).

Maintenance Demanded is derived from the Diagnosis ASE; see 7.3.4.4.4 for details.

Allowed Values: NO_MAINTENANCE_DEMANDED,
MAINTENANCE_DEMANDED

NOTE 8 If Shared Input contains the value SHARED_INPUT, this field contains always the value NO_MAINTENANCE_DEMANDED (see 7.4.6.2.2.1 for details).

Diag Info

This attribute indicates whether diagnosis information is present in at least one channel of the submodule (Format Indicator > V_1_0).

Diag Info is derived from the Diagnosis ASE; see 7.3.4.4.2 for details.

Allowed Values: NO_DIAG, DIAG

NOTE 9 If Shared Input contains the value SHARED_INPUT, this field contains always the value NO_DIAG (see 7.4.6.2.2.1 for details).

AR Info

This attribute contains the submodule state in relation to the established AR (Format Indicator > V_1_0).

AR Info is derived from the Ownership of the submodule; see 7.4.6.2.2 for details.

It shall be set according to Table 172.

Table 172 – AR Info

Value	Meaning
OWN	This AR is owner of the submodule
APPLICATION_READY_PENDING	This AR is owner of the submodule but the submodule is blocked by parameter checking pending; issued statically or dynamically. Static means that the submodule exists and a parameter fault was detected or the application of the submodule is not operating. Dynamic means that the submodule exists but the execution of the parameterization is not ready yet. In this case the submodule will issue a Return of Submodule alarm, when ready.

Value	Meaning
SUPERORDINATED_LOCKED	This AR is not owner of the submodule. It is blocked by superordinated means
LOCKED_BY_IO_CONTROLLER	This AR is not owner of the submodule. It is owned by another IOAR
LOCKED_BY_IO_SUPERVISOR	This AR is not owner of the submodule. It is owned by another IOSAR

NOTE 10 If Shared Input contains the value SHARED_INPUT, this field contains always the value OWN (see 7.4.6.2.2.1 for details).

Ident Info

This attribute contains the submodule state in respect to expected and real identification for an established AR (Format Indicator > V_1_0).

Ident Info is derived from the Real submodule; see 7.4.6.2.1 for details.

It shall be set according to Table 173.

Table 173 – Ident Info

Value	Meaning	Usable in conjunction with Module State "WRONG MODULE"
OK	No difference	No
SUBSTITUTE	Different but compatible	No
WRONG	Incompatible	Yes
NO	Empty	Yes

NOTE 11 The field Ident Info contains the value OK, if the real submodule is the expected one. The value SUBSTITUTE is returned, if the real submodule is able to substitute the expected one. The value WRONG is returned, if a real submodule is plugged, but doesn't match the expected one according to the type (ModuleID, SubmoduleID). The value NO is returned, if the subslot is empty, i.e. no real submodule is plugged. If Shared Input contains the value SHARED_INPUT, this field contains always the value OK (see 7.4.6.2.2.1 for details).

I&M Data

This attribute consists of the following attributes:

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This attribute defines to which module the I&M Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 7FFF

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the I&M Data Element belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

I&M0 Data

This attribute consists of the following attributes.

Vendor ID

This attribute defines the ID of the submodule's manufacturer. 0x0000 is used in case a manufacturer does not want to handle an ID.

Attribute type: Unsigned16

Allowed values: 0 to 0xFFFF

Order ID

This attribute defines the complete order number or at least a relevant part that allows unambiguous identification of the submodule within the manufacturer's order spectrum.

The attribute is filled as visible string. Unused octets shall be set to 0x20 (blank).

Attribute type: VisibleString[20]

Serial Number

This attribute defines the serial number of the submodule. The serial number is a unique production number of the device manufacturer even for devices with the same hardware, software or firmware edition.

The attribute is filled as visible string. Unused octets shall be set to 0x20 (blank).

Attribute type: VisibleString[16]

Hardware Revision

This attribute defines the hardware edition of the submodule.

Attribute type: Unsigned16

Allowed values: 0 to 0xFFFF

Software Revision

This attribute defines the software or firmware edition of the submodule. The structure supports coarse and detailed differentiation that may be defined by the manufacturer as "Vx.y.z". Any component that carries software shall present its software version even if the user may not be aware of it. V255.255.255 indicates the availability of profile specific information.

This attribute consists of the following attributes:

Prefix

The prefix attribute forms the "Vx.y.z" part of the software revision.

Attribute type: VisibleString[1]

Preferred values:

"V" (= officially released version) or "R" (= Revision)

"P" = Prototype

"U" = Under Test (field test)

"T" = Test Device

Functional Enhancement

The functional enhancement attribute forms the “Vx.y.z” part of the software revision.

Attribute type: Unsigned8

Allowed values: 0 to 0xFF

Bug Fix

The bug fix attribute forms the “Vx.y.z” part of the software revision.

Attribute type: Unsigned8

Allowed values: 0 to 0xFF

Internal Change

The internal change attribute forms the “Vx.y.z” part of the software revision.

Attribute type: Unsigned8

Allowed values: 0 to 0xFF

Revision Counter

This attribute defines the revision counter of the submodule.

A change of the revision counter marks a change of hardware or of its parameters. At production time the counter will be set to 0. This value is reserved for the first installation and the first increment. Thus, the counter increments from 1... 65 535, wrapping over back to 1 at the end.

The counter is incremented on submodule exchange or on any write access of parameters within the submodule and its activation.

To avoid unnecessary counter increments more tolerant and sophisticated criteria, meeting the minimum requirements of FDA (Food & Drug Administration) may be implemented manufacturer specifically (e.g. batch-ID, local parameter server, etc.).

The revision counter within the representative of the whole modular device – usually assigned to slot 0 – shall be incremented if any module's revision counter is incremented.

Attribute type: Unsigned16

Allowed values: 0 to 0xFFFF

Profile ID

This attribute defines the ID of the application profile the submodule is following. Submodules not following any PROFIBUS application profile shall use 0xF600.

Attribute type: Unsigned16

Allowed values: 0 to 0xFFFF

Profile Specific Type

In case a submodule follows a special application profile, this attribute offers information about the usage of its channels and/or sub devices. The submodule/channel information shall be according to the respective definitions of the application profile.

In case a submodule does not follow a special application profile, the attribute indicates the type of sensor or actuator.

Attribute type: Unsigned16

Allowed values: 0 to 0xFFFF

I&M Version

This attribute indicates the implemented version of the I&M functions.

Octet 1 (MSB) contains the major version number, e.g. 1 of version 1.0; octet 2 (LSB) contains the minor version number, e.g. 0 of version 1.0.

Attribute type: OctetString[2]

I&M Supported

This attribute indicates the availability of further I&M records. Bit 0 indicates the availability of Profile specific I&M, Bit 1 the availability of I&M1, Bit 2 the availability of I&M2, ... and Bit 15 the availability of I&M15.

Attribute type: Bit Sequence V2

I&M1 Data

This optional attribute consists of the following attributes.

Function

This attribute indicates the submodule's function or task. This may be a standard symbolic tag ("AKZ") out of a list or any other type of label defined by a configuration tool. Unused characters shall be set to 0x20 (blank).

Attribute type: VisibleString[32]

Default value: Filled with 0x20 (blank)

Location

This attribute indicates the submodule's location. This may be a standard symbolic tag ("OKZ") out of a list or any other type of label defined by a configuration tool. Unused characters shall be set to 0x20 (blank).

Attribute type: VisibleString[22]

Default value: Filled with 0x20 (blank)

I&M2 Data

This optional attribute consists of the following attribute.

Installation Date

This attribute indicates the date of installation or commissioning of a device or module. The content is a fixed date and time format according to ISO 8601: YYYY-MM-DD T HH:MM, where YYYY is the year in the usual Gregorian calendar, MM is the month of the year between 01 (January) and 12 (December), DD is the day of the month between 01 and 31, where HH is the hour of the day (0 - 23) and MM the minutes of the hour (0 - 59).

Time and date are normally separated by a "T" but for this appearance shall be separated by a blank character.

For example, the fourth day of February in the year 1995 is written in the standard notation as:

1995-02-04 16:23

Unused characters shall be set to 0x20 (blank).

Attribute type: VisibleString[16]

Default value: Filled with 0x20 (blank)

I&M3 Data

This optional attribute consists of the following attributes.

Descriptor

This attribute allows storing any individual additional information and annotation.

Unused characters shall be set to 0x20 (blank).

Attribute type: VisibleString[54]

Default value: Filled with 0x20 (blank)

I&M4 Data

This optional attribute consists of the following attributes.

Signature

This attribute allows parameterization tools to store a "security" code as a reference for a particular parameterization session and audit trail tools to retrieve the code for integrity checks. Together with the "asset identification" consisting of Vendor ID, Order ID, and Serial Number the Signature allows unambiguousness of the session data.

Unused characters shall be set to 0x00.

Attribute type: OctetString[54]

Default value: Filled with 0x00

I&M Filter Data

This attribute consists of the following attributes:

I&M Carrier Data

This attribute contains all submodules being carrier of identification and maintenance data (see 7.1.4.5.7 for the definition).

It consists of the following attributes:

List of APs

One API is composed of the following list elements:

API

This attribute defines the number of the application process.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This attribute defines to which module the I&M carrier belongs.

Attribute type: Unsigned16

Allowed values: 0 to 7FFF

Module Ident Number

This attribute defines the identification of the slot.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFFFFFF

NOTE 12 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the I&M carrier belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

Submodule Ident Number

This attribute defines the identification of the subslot.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

NOTE 13 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Module Representative Data

This optional attribute indicates the submodules acting as module representative and thus carrying the module's I&M data. It consists of the following attributes.

NOTE 14 The attribute indicates at most one submodule per module acting as module representative.

List of APs

One API is composed of the following list elements:

API

This attribute defines the number of the application process.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This attribute defines to which module the module representative belongs.

Attribute type: Unsigned16

Allowed values: 0 to 7FFF

Module Ident Number

This attribute defines the identification of the slot.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFFFFFF

NOTE 15 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the module representative belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

Submodule Ident Number

This attribute defines the identification of the subslot.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

NOTE 16 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Device Representative Data

This attribute contains one submodule, which is a carrier of identification and maintenance information (see 7.1.4.5.7 for the definition) and provides access to I&M1, I&M2, I&M3 and I&M4.

It consists of the following attributes.

NOTE 17 The attribute indicates at most one submodule acting as device representative.

List of APs

One API is composed of the following list elements:

API

This attribute defines the number of the application process.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This attribute defines to which module the device representative belongs.

Attribute type: Unsigned16

Allowed values: 0 to 9FFF

Module Ident Number

This attribute defines the identification of the slot.

Attribute type: Unsigned32

Allowed values: 1 to 0xFFFFFFFF

NOTE 18 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This attribute defines to which subslot the device representative belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

Submodule Ident Number

This attribute defines the identification of the subslot.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

NOTE 19 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Autoconfiguration Data

This optional attribute allows establishing an application relation to an IO device without the need of an engineering system. For this the IO device returns information about its communication capabilities and lists its submodules which are able to be used without the need for startup parameterization.

It consists of the following attributes:

Communication Data

This attribute defines the communication capabilities. It consists of the following attributes:

Min Device Interval

This attribute indicates the minimum device interval supported by the IO device in multiples of 31,25 µs.

Attribute type: Unsigned16

Allowed Values: 0x1 to 0x1000

Time base: 31,25 µs

Communication Properties

This attribute indicates the communication properties of the IO device. It consists of the following attributes:

DFP

This attribute indicates, whether the IO device is capable for dynamic frame packing. If the attribute is set to TRUE, the IO device shall support RT_CLASS_3.

Attribute type: Boolean

RTC3

This attribute indicates, whether the IO device supports RT_CLASS_3.

Attribute type: Unsigned8

Allowed Values:

- 0x00 (RT Class 3 not supported),
- 0x01 (RT Class 3 supported, PD IR Frame Data needed),
- 0x02 (RT Class 3 supported, PD IR Frame Data not supported),
- 0x03 (RT Class 3 supported, operation with and without PD IR Frame Data supported)

Send Clock Factors

This attribute contains the send clocks supported by the IO device. The attribute is interpreted as a series of bits. Each bit of the array corresponds with one Send Clock Factor and shall be calculated as follows

$$N = (\text{Send Clock Factor} - 1) \text{ DIV } 8$$

$$M = (\text{Send Clock Factor} - 1) \text{ MOD } 8$$

where

N is the number of the octet, and

M is the number of bit within the octet.

The send clock itself is computed by multiplying the Send Clock Factor with 31,25 µs. An individual bit shall be set to TRUE, if the send clock factor is supported.

Attribute type: OctetString[16]

Configuration Data

This attribute contains autoconfiguration information related to the configuration. It contains those submodules of the IO device usable for operation without any parameterization (i.e. the submodule is operable with a suitable default). If a submodule is not operable without parameterization, it shall not be part of this list.

The configuration data consists of the following attribute:

List of Offered Submodules

This attribute contains the Configuration Data Elements. A list element describes a particular submodule usable without parameterization and consists of the following attributes:

API

This attribute contains the application profile index of the application profile the submodule belongs to.

Attribute type: Unsigned32

Allowed Values: 0x00000000 to 0xFFFFFFFF

Slot Number

This attribute contains the slot number containing the submodule.

Attribute type: Unsigned16

Allowed Values: 0x0000 to 0x7FFF

Subslot Number

This attribute contains the subslot number containing the submodule.

Attribute type: Unsigned16

Allowed Values: 0x0000 to 0x7FFF

Module Ident Number

This attribute contains the module identification of the slot containing the submodule.

Attribute type: Unsigned32

Allowed Values: 0x1 to 0xFFFFFFFF

NOTE 19 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Submodule Ident Number

This attribute contains the submodule identification.

Attribute type: Unsigned32

Allowed Values: 0x1 to 0xFFFFFFFF

NOTE 20 The attribute value is manufacturer specific. See 7.1.4.5.5 for details.

Length Input Data

This attribute contains the number of octets of the submodule's input data element without IOPS.

Attribute type: Unsigned16

Allowed Values: 0 to 1 439

Length Output Data

This attribute contains the number of octets of the submodule's output data element without IOPS.

Attribute type: Unsigned16

Allowed Values: 0 to 1 439

Isochronous Data

This optional attribute contains autoconfiguration information related to isochronously operable submodule. It contains those submodules of the IO device usable for isochronous operation without any parameterization (i.e. the submodule is operable with a suitable default) besides the isochronous parameters. If a submodule is not isochronously operable without parameterization, it shall not be part of this list.

Each submodule offered in the isochronous data has to be part also of the configuration data. Thus module ident number, submodule ident number and length for input and output data are used from the appropriate entry in the configuration data.

The isochronous data consists of the following attribute:

List of Offered Submodules

This attribute contains the Isochronous Data Elements. A list element describes a particular submodule usable without parameterization and consists of the following attributes:

API

This attribute contains the application profile index of the application profile the submodule belongs to.

Attribute type: Unsigned32

Allowed Values: 0x00000000 to 0xFFFFFFFF

Slot Number

This attribute contains the slot number containing the submodule.

Attribute type: Unsigned16

Allowed Values: 0x0000 to 0x7FFF

Subslot Number

This attribute contains the subslot number containing the submodule.

Attribute type: Unsigned16

Allowed Values: 0x0000 to 0x7FFF

Time IO Base (T_IO_Base)

This attribute determines the granularity for input and output delay time adjustment.

The time for T_IO_Base is calculated by $T\_IO\_Base * 1 \text{ ns}$.

Time base: 1 ns

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time IO Input (T_IO_Input)

This attribute contains the factor for a possible input refresh delay time. It may be the smallest possible time to copy or preprocess the input data.

The time for T_IO_Input is calculated by $T\_IO\_Input * T\_IO\_Base * 1 \text{ ns}$.

Time base: $T\_IO\_Base * 1 \text{ ns}$

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time IO Output (T_IO_Output)

This attribute contains the factor for a possible output refresh delay time. It may be the smallest possible time to copy or preprocess the output data.

The time for T_IO_Output is calculated by $T\_IO\_Output * T\_IO\_Base * 1 \text{ ns}$.

Time base: $T\_IO\_Base * 1 \text{ ns}$

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Fast Startup Data

This optional attribute allows adjusting the fast startup behavior per individual application relation

It consists of the following attributes:

List of ARs

One AR is composed of the following list elements:

AREP

This parameter is the local identifier for the desired AR.

Expected Fast Startup Data

This attribute contains the expected Fast Startup Parameter Data for this AR. It is composed of the following attributes:

FS Parameter Block

This optional attribute contains the Fast Startup Parameter Data for the AR. It is composed of the following attributes:

FS Parameter Mode

This attribute contains the currently active Parameter Mode of the AR.

Attribute type: Unsigned32

Allowed values: ON, OFF

FS Parameter UUID

This attribute contains a UUID given by an engineering system to indicate unchanged parameter sets.

Attribute type: UUID

Manufacturer Specific Fast Startup Data

This optional attribute shall contain manufacturer specific fast startup data.

Attribute type: OctetString

GSD Data

This optional attribute hosts the GSD for this IO device.

Attribute type: OctetString

7.3.6.2.3 Invocation of the Context object

Each IO Device shall invoke one IO Device Context object.

A context service request (i.e. connect, prm begin...) shall be answered within less than 300 s; otherwise the client of the service may assume that the service request timed out.

7.3.6.2.4 Binding of the Context services

The context ASE's services are bound to lower level services as following, shown in Table 174:

Table 174 – Binding Context services

Service	Binding	Usage
Connect	bound to RPC::Connect	used by IO controller or IO supervisor
Read	bound to Record Data::Read	used by IO controller or IO supervisor
Write	bound to Record Data::Write	used by IO controller or IO supervisor
Control	bound to RPC::Control	used by IO controller or IO supervisor or IO device
Release	bound to RPC::Release	used by IO controller or IO supervisor

7.3.6.3 Context service specification

7.3.6.3.1 Connect

7.3.6.3.1.1 General

Two types of connect service are supported:

- The Connect service (see 7.3.6.3.1.2) to establish an application relation for the use of an IO controller or an IO supervisor with status and control functions.
- The Connect Device Access service (see 7.3.6.3.1.3) to establish an application relation for the use of an IO supervisor without status and control functions.

7.3.6.3.1.2 Connect

The Connect service is used to establish an IO AR or Supervisor AR. The application process shall check the service parameter of the request/indication with the local parameter in its ASE objects (Real Identification attributes). In case of a service parameter mismatch (e.g. submodules are already used by other IO ARs) with the object attributes the application process shall respond the mismatching modules within response. The client may take the appropriate actions. If the establishment phase was successfully the service parameter are stored as attribute values in the Expected Identification attribute group of the IO Device Context object.

Table 175 shows the parameters of the service.

NOTE 1 The service parameter IP Address and Object UUID are necessary to address the destination device. The underlying model assumes that this information is provided by the AR information, which has to be loaded before on behalf of the client application. On server side this information is also handled inside the application layer and does not appear as an explicit service parameter. Local implementation may pass this service parameter to the application process. The same model applies to Read services over the implicit AR.

The application of the server receiving the Connect indication service primitive shall check the service parameter. Especially, the parameter Module Ident Number, Submodule Ident Number, Submodule Data Length, IOCS Length, and IOPS Length shall only be checked by the user application.

The resources for an application relation are allocated with the Connect of the application relation. This includes the resources for the application relation itself and the resources of the IOCRs. The resources are freed when the application relation is terminated.

For the first application relation having ARType IOCARSR and belonging to a new AR Set (as indicated by the ARUUID; details see IEC 61158-6-10) the resources of the complete AR Set are allocated; i.e. for all ARs of the complete AR Set. The resources of an application relation belonging to the same AR Set established at some later point are assigned from this allocated

memory. The resources of the AR Set are freed when the last application relation of the AR Set is terminated.

If the application relation cannot be established due to scarce resources, the Connect is rejected with an “out of AR/ARset resources” error.

If an application relation shall contain the PDev submodules which are already owned by another application relation, the Connect is rejected with an “PDev already owned” error.

The send clock factors of all application relations established in parallel to a particular IO device have to be the same. Thus the send clock factor of the IO device is determined by the application relation first established. The Connect of a succeeding application relation having a different send clock factor is rejected with an error for the corresponding block / parameter.

The IRDataUUID of all application relations using RT_CLASS_3 established in parallel to a particular IO device have to be the same. The IRDataUUID of the IO device is determined by the application relation first established with RT_CLASS_3. The Connect of a succeeding application relation using RT_CLASS_3 having a different IRDataUUID is rejected with an error for the corresponding block / parameter.

Table 175 – Connect

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
AR Parameter Block	M	M(=)		
AR type	M	M(=)		
AR UUID	M	M(=)		
Session Key	M	M(=)		
CM Initiator Mac Add	M	M(=)		
CM Initiator Object UUID	M	M(=)		
AR Properties	M	M(=)		
CM Initiator Activity Timeout Factor	M	M(=)		
Initiator UDP RT Port	M	M(=)		
CM Initiator Station Name	M	M(=)		
List of IO CR Parameter Blocks	M	M(=)		
IO CR type	M	M(=)		
IO CR Reference	M	M(=)		
LT Field	M	M(=)		
IO CR Properties	M	M(=)		
C_SDU Length	M	M(=)		
Frame ID	M	M(=)		
Send Clock Factor	M	M(=)		
Reduction Ratio	M	M(=)		
Phase	M	M(=)		
Sequence	M	M(=)		
Frame Send Offset	M	M(=)		
Data Hold Factor	M	M(=)		
IO CR Tag Header	M	M(=)		
IO CR Multicast MAC Add	M	M(=)		
List Of APIs	M	M(=)		

Parameter name	req	ind	rsp	cnf
API	M	M(=)		
List of Related IO Data Objects	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
IO Data Object Frame Offset	M	M(=)		
List of Related IOCS	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
IOCS Frame Offset	M	M(=)		
Alarm CR Parameter Block	M	M(=)		
Alarm CR type	M	M(=)		
LT Field	M	M(=)		
Alarm CR Properties	M	M(=)		
RTA Timeout Factor	M	M(=)		
RTA Retries	M	M(=)		
Local Alarm Reference	M	M(=)		
Max Alarm Data Length	M	M(=)		
Alarm CR Tag Header High	M	M(=)		
Alarm CR Tag Header Low	M	M(=)		
List of Expected Submodule Blocks	U	U(=)		
List of Related APIs	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Module Ident Number	M	M(=)		
Module Properties	M	M(=)		
List of Submodules	M	M(=)		
Subslot Number	M	M(=)		
Submodule Ident Number	M	M(=)		
Submodule Properties	M	M(=)		
List of Data Descriptions	M	M(=)		
Data Description	M	M(=)		
Submodule Data Length	M	M(=)		
Length IOPS	M	M(=)		
Length IOCS	M	M(=)		
Parameter Server Block	U	U(=)		
Parameter Server Object UUID	M	M(=)		
Parameter Server Properties	M	M(=)		
CM Initiator Activity Timeout Factor	M	M(=)		
Parameter Server Station Name	M	M(=)		
List of Multicast CR Blocks	U	U(=)		
IO CR Reference	M	M(=)		
Address Resolution Properties	M	M(=)		
MCI Timeout Factor	M	M(=)		
Provider Station Name	M	M(=)		
AR RPC Block	U	U(=)		

Parameter name	req	ind	rsp	cnf
Initiator RPC Server Port	M	M(=)		
IR Info Block	U	U(=)		
IR Data UUID	M	M(=)		
Number of IO CRs	M	M(=)		
List of Sub Frame Data	U	U(=)		
IO CR Reference	M	M(=)		
Sub Frame Offset	M	M(=)		
Sub Frame Data	M	M(=)		
SR Info Block	U	U(=)		
Redundancy Data Hold Factor	M	M(=)		
SR Properties	M	M(=)		
Input Valid on Backup AR	M	M(=)		
Activate Redundancy Alarm	M	M(=)		
AR Server Block	U	U(=)		
CM Initiator Station Name	M	M(=)		
List of AR Vendor Block Requests	U	U(=)		
AP Structure Identifier	M	M(=)		
API	M	M(=)		
Data Request	M	M(=)		
AR FSU Block	U	U(=)		
FS Parameter Block	U	U(=)		
FS Parameter Mode	M	M(=)		
FS Parameter UUID	M	M(=)		
Fast StartUp Block	U	U(=)		
Data	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
AR Response Block			M	M(=)
AR type			M	M(=)
AR UUID			M	M(=)
Session Key			M	M(=)
CM Responder MAC Add			M	M(=)
Responder UDP RT Port			M	M(=)
List of IO CR Response Blocks			M	M(=)
IO CR type			M	M(=)
IO CR Reference			M	M(=)
Frame ID			M	M(=)
Alarm CR Response Block			M	M(=)
Alarm CR type			M	M(=)
Local Alarm Reference			M	M(=)
Max Alarm Data Length			M	M(=)
Module Diff Block			C	C(=)
List of APIs			M	M(=)
API			M	M(=)

Parameter name	req	ind	rsp	cnf
List of Modules			M	M(=)
Slot Number			M	M(=)
Module Ident Number			M	M(=)
Module State			M	M(=)
List of Submodules			M	M(=)
Subslot Number			M	M(=)
Submodule Ident Number			M	M(=)
Submodule State			M	M(=)
AR RPC Block			C	C(=)
Responder RPC Server Port			M	M(=)
List of AR Vendor Block Responses			C	C(=)
AP Structure Identifier			M	M(=)
API			M	M(=)
Data Response			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

AR Parameter Block

The parameter contains of the following attributes:

AR type

This parameter contains the value for the equivalent attribute of the ARL class specification.

AR UUID

This parameter contains the value for the equivalent attribute of the ARL class specification.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 2 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Attribute type: Unsigned16

CM Initiator MAC Add

This parameter contains the value for the equivalent attribute of the ARL class specification.

IO Initiator Object UUID

This parameter contains the value for the equivalent attribute of the ARL class specification.

AR Properties

This parameter contains the value for the equivalent attribute of the ARL class specification.

CM Initiator Activity Timeout Factor

This parameter contains the value for the equivalent attribute of the ARL class specification.

Initiator UDP RT Port

This parameter contains the value for the equivalent attribute of the ARL class specification.

CM Initiator Station Name

This parameter contains the value for the equivalent attribute of the ARL class specification.

List of IO CR Parameter Blocks

The parameter list contains the following attributes:

IO CR type

This parameter contains the value for the equivalent attribute of the CRL class specification.

IO CR Reference

This parameter contains the value to identify the CR within this service.

NOTE 3 It is used within the context of the connect service to link the IO Data elements with its CRs. In the local context of the IO device it is transferred into the CREP.

LT Field

This parameter contains the value for the equivalent attribute of the CRL class specification.

IO CR Properties

This parameter contains the value for the equivalent attribute of the CRL class specification.

C_SDU Length

This parameter contains the value for the equivalent attribute of the CRL class specification.

Frame ID

This parameter contains the value for the equivalent attribute of the CRL class specification.

Send Clock Factor

This parameter contains the value for the equivalent attribute of the CRL class specification.

Reduction Ratio

This parameter contains the value for the equivalent attribute of the CRL class specification.

Phase

This parameter contains the value for the equivalent attribute of the CRL class specification.

Sequence

This parameter contains the value for the equivalent attribute of the CRL class specification.

Frame Send Offset

This parameter contains the value for the equivalent attribute of the CRL class specification.

Data Hold Factor

This parameter contains the value for the equivalent attribute of the CRL class specification.

IO CR Tag Header

This parameter contains the value for the equivalent attribute of the CRL class specification.

IO CR Multicast MAC Add

This parameter contains the value for the equivalent attribute of the CRL class specification.

List Of APIs

This parameter contains the value for the equivalent attribute of the CRL class specification.

API

This parameter contains the value for the equivalent attribute of the CRL class specification.

List of Related IO Data Objects

The parameter list shall always be present. It contains the slots and subslots where read and write access is allowed. The application process of the server shall respond to write services addressed to objects beyond this negotiated address space with a negative response. Read access is always allowed to other objects. The following parameter shall be checked with the corresponding attributes of the IO Data object and the result in case of a positive response shall be stored as attribute values of the Context object.

The user of the service should not specify the same IO Data object multiple times within an IOCR, since this may not be supported by the IO device.

The parameter consists of the following sub-parameter:

Slot Number

This parameter contains the value for the equivalent attribute of the Context ASE.

Subslot Number

This parameter contains the value for the equivalent attribute of the Context ASE.

IO Data Object Frame Offset

This parameter contains the value for the equivalent attribute of the Context ASE.

List of Related IOCS

This parameter contains the value for the equivalent attribute of the Context ASE.

Slot Number

This parameter contains the value for the equivalent attribute of the Context ASE.

Subslot Number

This parameter contains the value for the equivalent attribute of the Context ASE.

IOCS Frame Offset

This parameter contains the value for the equivalent attribute of the Context ASE.

Alarm CR Parameter Block

This parameter consists of the following attributes:

Alarm CR type

This parameter contains the value for the equivalent attribute of the CRL class specification.

LT Field

This parameter contains the value for the equivalent attribute of the CRL class specification.

Alarm CR Properties

This parameter contains the value for the equivalent attribute of the CRL class specification.

RTA Timeout Factor

This parameter contains the value for the equivalent attribute of the CRL class specification.

RTA Retries

This parameter contains the value for the equivalent attribute of the CRL class specification.

Local Alarm Reference

This parameter contains the value for the equivalent attribute of the CRL class specification.

Max Alarm Data Length

This parameter contains the value for the equivalent attribute of the CRL class specification.

Alarm CR Tag Header High

This parameter contains the value for the equivalent attribute of the CRL class specification.

Alarm CR Tag Header Low

This parameter contains the value for the equivalent attribute of the CRL class specification.

List of Expected Submodule Blocks

This parameter list consists of the following parameter:

NOTE 4 The list includes only "real" submodules but does not include "empty" slots.

List Of Related APIs

This parameter contains the value for the equivalent attribute of the CRL class specification.

API

This parameter contains the value for the equivalent attribute of the CRL class specification.

Slot Number

This parameter contains the value for the equivalent attribute of the Context ASE.

Module Ident Number

This parameter contains the value for the equivalent attribute of the Context ASE.

Module Properties

This parameter contains the value for the equivalent attribute of the Context ASE.

List of Submodules

This list consists of the following attributes:

Subslot Number

This parameter contains the value for the equivalent attribute of the Context ASE.

Submodule Ident Number

This parameter contains the value for the equivalent attribute of the Context ASE.

Submodule Properties

This parameter contains the value for the equivalent attribute of the Context ASE.

List of Data Descriptions

This list consists of the following attributes:

Data Description

This parameter contains the value for the equivalent attribute of the Context ASE.

Submodule Data Length

This parameter contains the value for the equivalent attribute of the Context ASE.

Length IOPS

This parameter contains the value for the equivalent attribute of the Context ASE.

Length IOCS

This parameter contains the value for the equivalent attribute of the Context ASE.

Parameter Server Block

This parameter consists of the following attributes:

Parameter Server Object UUID

This parameter contains the value for the equivalent attribute of the ASE.

Parameter Server Properties

This parameter contains the value for the equivalent attribute of the ASE.

CM Initiator Activity Timeout Factor

This parameter contains the value for the equivalent attribute of the ASE.

Parameter Server Station Name

This parameter contains the value for the equivalent attribute of the ASE.

List of Multicast CR Block

This parameter consists of the following attributes:

IO CR Reference

This parameter contains the value for the equivalent attribute of the AR ASE.

Address Resolution Properties

This parameter contains the value for the equivalent attribute of the AR ASE.

MCI Timeout Factor

This parameter contains the value for the equivalent attribute of the AR ASE.

Provider Station Name

This parameter contains the value for the equivalent attribute of the AR ASE.

AR RPC Block

This optional parameter shall be used by the IO device to address the Initiator RPC Server Port instead of the port of the endpoint mapper. It consists of the following attributes:

Initiator RPC Server Port

This parameter contains the value for the equivalent attribute of the ASE.

IR Info Block

This optional parameter shall be used by the IO device to determine whether the application relation references to IR data, which is part of the persistent PDev parameterization. The parameter shall be used only when using RT_CLASS_3 with advanced startup mode. The parameter allows multiple IO controllers sharing the IO device using RT_CLASS_3. Only one of the IO controllers is able to parameterize the IR data at the PDev, the others reference the IR data configured by that IO controller via the IR Info Block. It consists of the following attributes:

IR Data UUID

This parameter contains the value for the equivalent attribute of the AR ASE.

Number of IO CRs

This parameter contains the value for the equivalent attribute of the AR ASE.

List of DFP IO CRs

This parameter consists of the following attributes:

IO CR Reference

This parameter contains the value for the equivalent attribute of the CRL ASE.

Sub Frame Offset

This parameter contains the value for the equivalent attribute of the AR ASE.

Sub Frame Data

This parameter consists of the following attributes:

Sub Frame Position

This parameter contains the value for the equivalent attribute of the AR ASE.

Sub Frame Data Length

This parameter contains the value for the equivalent attribute of the AR ASE.

SR Info Block

This optional attribute defines the system redundancy parameters of the desired AR. This attribute is present only if the startup mode of the AR is advanced. It consists of the following attributes:

Redundancy Data Hold Factor

This parameter contains the value for the equivalent attribute of the AR ASE.

SR Properties

This attribute is used to define system redundancy behavior. It consists of the following attributes:

Input Valid on Backup AR

This parameter contains the value for the equivalent attribute of the AR ASE.

Activate Redundancy Alarm

This parameter contains the value for the equivalent attribute of the AR ASE.

AR Server Block

This optional attribute is present only if the startup mode of the AR is advanced. It allows to check, whether the device responding to the connect service is the expected one by checking its name. It consists of the following attributes:

CM Responder Station Name

This parameter contains the name of the CM responder.

NOTE 5 The motivation for this block is the phase between a DCP::Identify(name) and the subsequent connect to the IP address of the device. If in-between the name of the device has been modified e.g. by DCP::Set, the connect would not detect this situation. With this block the IO controller is able to check for name == CM responder station name. If unequal, the IO controller would then abort the connect.

List of AR Vendor Block Requests

This optional attribute allows transferring normative, profile- or vendor specific parameters for the desired AR. This attribute is present only if the startup mode of the AR is advanced.

Any possible combination of API and APStructureIdentifier may exist only once in this list.

An IO device shall not expect an ordering within this list.

It consists of the following attributes:

AP Structure Identifier

This parameter contains the value for the equivalent attribute of the AR ASE.

API

This parameter contains the value for the equivalent attribute of the AR ASE.

Data Request

This parameter contains the value for the equivalent attribute of the AR ASE.

AR FSU Block

This optional attribute allows to transfer fast startup specific parameters for the desired AR. This attribute is present only if the startup mode of the AR is advanced. It consists of the following attributes:

FS Parameter Block

This optional parameter consists of the following elements:

FS Parameter Mode

This parameter consists of the following elements:

Mode

Mode is used to establish the fast startup mode of the application.

Attribute type: Unsigned8

Allowed values: OFF (0x00), ON (0x01)

Default value: OFF

FS Parameter UUID

This parameter contains a UUID indicating the generation of the record data configuration for the device. The UUID is changed by engineering whenever something within the record data changes, records are added or subtracted from the configuration.

Attribute type: UUID

Fast Startup Block

This optional parameter consists of the following elements:

Data

This parameter holds data.

Attribute type: OctetString

Result(+)

This parameter indicates that the service request succeeded. It shall only contain IO Data Response Block parameter for entries that differ from the requested parameter values.

AR Response Block

The selectable parameter contains of the following attributes:

AR type

This parameter contains the value for the equivalent attribute of the ARL class specification.

AR UUID

This parameter contains the value for the equivalent attribute of the ARL class specification.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 6 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Attribute type: Unsigned16

CM Responder MAC Add

This parameter contains the value for the equivalent attribute of the ARL class specification.

Responder UDP RT Port

This parameter contains the value for the equivalent attribute of the ARL class specification.

List of IO CR Response Blocks

The parameter list contains the following attributes:

IO CR Type

This parameter contains the value for the equivalent attribute of the CRL class specification.

IO CR Reference

This parameter contains the value mirrored from the request.

Frame ID

This parameter contains the value for the equivalent attribute of the CRL class specification.

Alarm CR Response Block

This parameter consists of the following attributes:

Alarm CR type

This parameter contains the value for the equivalent attribute of the CRL class specification.

Local Alarm Reference

This parameter contains the value for the equivalent attribute of the CRL class specification.

Max Alarm Data Length

This parameter contains the value for the equivalent attribute of the CRL class specification.

Module Diff Block

The Module Diff Block contains those submodules which differ from the requested ones; that are those submodules not matching the expected configuration (wrong, missing or substituted) and submodules not being part of the application relation (superordinated locked, locked by other IO controller or locked by supervisor), and submodules having diagnosis and/or maintenance.

The parameter consists of the following sub-parameters:

List of APIs

This parameter contains the APIs which differ from the requested configuration.

API

This parameter contains the value for the equivalent attribute of the ASE.

List of Modules

This parameter contains the modules which differ from the requested configuration.

Slot Number

This parameter contains the value for the equivalent attribute of the ASE.

Module Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Module State

This parameter contains the value for the equivalent attribute of the ASE.

List of Submodules

This parameter list consists of the following parameter:

Subslot Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule State

This parameter contains the value for the equivalent attribute of the ASE.

AR RPC Block

This optional parameter shall be used by the initiator to address the Responder RPC Server Port instead of the port of the endpoint mapper. It consists of the following attributes.

Responder RPC Server Port

This parameter contains the value for the equivalent attribute of the ASE.

NOTE 7 It is necessary to query the endpoint mapper to get this port to avoid firewall problems.

List of AR Vendor Block Response

This optional attribute allows to transfer the response normative, profile- or vendor specific parameters for the desired AR. This attribute is present only if the startup mode of the AR is advanced and if an AR Vendor Block Request has been used.

For every AR Vendor Block request this list shall contain a vendor block response in the same ordering as in the request. The response shall not contain additional blocks.

It consists of the following attributes:

AP Structure Identifier

This parameter contains the value for the equivalent attribute of the AR ASE.

API

This parameter contains the value for the equivalent attribute of the AR ASE.

Data Response

This parameter contains the value for the equivalent attribute of the AR ASE.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter shall have the value PNIO.

Type: Unsigned8

Error code 1

The Error code 1 assumes one of the following values:
FAULTY_CONNECT_BLOCK.

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 8 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 9 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.6.3.1.3 Connect Device Access

The Connect service is used to establish a Supervisor AR for device access. The application process shall check the service parameter of the request/indication with the local parameter in its ASE objects (Real Identification attributes). The client may take the appropriate actions. If the establishment phase was successfully the service parameter are stored as attribute values in the Expected Identification attribute group of the IO Device Context object.

Table 176 shows the parameters of the service.

NOTE 1 The service parameter IP Address and Object UUID are necessary to address the destination device. The underlying model assumes that this information is provided by the AR information, which has to be loaded before on behalf of the client application. On server side this information is also handled inside the application layer and does not appear as an explicit service parameter. Local implementation may pass this service parameter to the application process. The same model applies to Read services over the implicit AR.

The application of the server receiving the Connect indication service primitive shall check the service parameter.

Table 176 – Connect Device Access

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		

Parameter name	req	ind	rsp	cnf
AR Parameter Block	M	M(=)		
AR type	M	M(=)		
AR UUID	M	M(=)		
Session Key	M	M(=)		
CM Initiator MAC Add	M	M(=)		
CM Initiator Object UUID	M	M(=)		
AR Properties	M	M(=)		
CM Initiator Activity Timeout Factor	M	M(=)		
Initiator UDP RT Port	M	M(=)		
CM Initiator Station Name	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
AR Response Block			M	M(=)
AR type			M	M(=)
AR UUID			M	M(=)
Session Key			M	M(=)
CM Responder MAC Add			M	M(=)
Responder UDP RT Port			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

AR Parameter Block

The parameter contains of the following attributes:

AR type

This parameter contains the value for the equivalent attribute of the ARL class specification.

AR UUID

This parameter contains the value for the equivalent attribute of the ARL class specification.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 2 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Attribute type: Unsigned16

CM Initiator MAC Add

This parameter contains the value for the equivalent attribute of the ARL class specification.

IO Initiator Object UUID

This parameter contains the value for the equivalent attribute of the ARL class specification.

AR Properties

This parameter contains the value for the equivalent attribute of the ARL class specification.

CM Initiator Activity Timeout Factor

This parameter contains the value for the equivalent attribute of the ARL class specification.

Initiator UDP RT Port

This parameter contains the value for the equivalent attribute of the ARL class specification.

CM Initiator Station Name

This parameter contains the value for the equivalent attribute of the ARL class specification.

Result(+)

This parameter indicates that the service request succeeded. It shall only contain IO Data Response Block parameter for entries that differ from the requested parameter values.

AR Response Block

The selectable parameter contains of the following attributes:

AR type

This parameter contains the value for the equivalent attribute of the ARL class specification.

AR UUID

This parameter contains the value for the equivalent attribute of the ARL class specification.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 3 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Attribute type: Unsigned16

CM Responder MAC Add

This parameter contains the value for the equivalent attribute of the ARL class specification.

Responder UDP RT Port

This parameter contains the value for the equivalent attribute of the ARL class specification.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter shall have the value PNIO.

Type: Unsigned8

Error code 1

The Error code 1 assumes one of the following values:
FAULTY_CONNECT_BLOCK.

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 4 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 5 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.6.3.2 Release

This service shall be used by an IO controller or IO supervisor to disconnect an IO or Supervisor AR.

Table 177 shows the parameters of the service.

Table 177 – Release

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Session Key	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Session Key			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 1 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter shall contain the value PNIO.

Type: Unsigned8

Error code 1

The Error code 1 assumes one of the following values:
FAULTY_RELEASE_BLOCK

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 2 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 3 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.6.3.3 Abort

This service shall be used by an IO device to close an IO or Supervisor AR in case of a serious error when operation is no longer possible. Table 178 shows the parameters of the service.

Table 178 – Abort

Parameter name	req
Argument	M
AREP	M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

7.3.6.3.4 Local AR Abort

This service shall be used by an IO device or an IO or Supervisor AR to indicate a local abort of an existing AR. Table 179 shows the parameters of the service.

Table 179 – Local AR Abort

Parameter name	req	ind	cnf
Argument	M(=)	M(=)	M(=)
AREP	M(=)	M(=)	M(=)
Session Key	M(=)	M(=)	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Attribute type: Unsigned16

7.3.6.3.5 Prm Begin

This Service allows an acknowledged parameterization in the context of an established AR.

The service Prm Begin is oriented one-to-one and onto an application relation indicated via the parameter AREP. Issuing the service Prm Begin is allowed at any time after the IN-DATA notification for the desired AR (after the Application Ready.rsp). The service Prm Begin allows starting an acknowledged re-parameterization sequence for the submodules indicated via the List of Submodules.

All submodules noted are parameterized within the parameterization sequence; during the parameterization sequence the IO controller delivers a consistent set of parameters by issuing IODWrite.req or else IODWriteMultiple.req services. The IO controller completes parameter delivery by issuing the Prm End service. The absence of parameters for a submodule to be parameterized means that the submodule shall use the default parameters.

The parameterization sequence is finished by the IO device issuing Application Ready to the IO controller, again narrowed onto the submodules contained in the parameterization sequence and containing a ModuleDiffBlock oriented to the submodules indicated in the request. After this Application Ready the submodules are again usable for the IO controller.

For the IOxS of the affected submodules the rules for the “normal” startup parameterization sequence apply: They are set to “Good” by the IO device as soon as the parameterization of the appropriate submodule is finished and the submodule is able to deliver or receive data according to the new parameters.

The list of submodules shall have the same or a subset of those submodules listed, as the expected submodule list within the Connect service for the desired AR. If in the list are more or different submodules than are present in the expected submodule List of the AR-Connect, then the device sees a protocol error and aborts the AR.

After a device receives a Prm Begin signal, an affected submodule shall not issue an alarm until the corresponding Application Ready; it shall store those alarms and issue them after the corresponding Application Ready.

Already issued Alarms except Plug and Release shall be acknowledged by the IO controller. To get into a consistent state the IO device waits with its Application Ready until all outstanding alarms are acknowledged.

Plug and Release shall be stored by the IO controller. They shall be processed and acknowledged after the Application Ready.

If a device does not support the service Prm Begin, then it shall acknowledge with protocol error and abort the AR.

Table 180 shows the parameters of the service.

Table 180 – Prm Begin

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Session Key	M	M(=)		
List of Submodules	U	U(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Session Key			M	M(=)
Module Diff Block			C	C(=)
List of APIs			M	M(=)
API			M	M(=)
List of Modules			M	M(=)
Slot Number			M	M(=)
Module Ident Number			M	M(=)
Module State			M	M(=)
List of Submodules			M	M(=)
Subslot Number			M	M(=)
Submodule Ident Number			M	M(=)
Submodule State			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)

Parameter name	req	ind	rsp	cnf
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 1 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

List of Submodules

This optional parameter contains the list of submodules which shall be reparameterized.

API

This parameter contains the value for the equivalent attribute of the ASE.

Slot Number

This parameter contains the value for the equivalent attribute of the ASE.

Subslot Number

This parameter contains the value for the equivalent attribute of the ASE.

Result(+)

This parameter indicates that the service request succeeded.

Module Diff Block

The Module Diff Block contains those submodules not matching the expected configuration (wrong, missing or substituted) and submodules not being part of the application relation (superordinated locked, locked by other IO controller or locked by supervisor). Optionally submodules having diagnosis and/or maintenance are stated also in the list.

The parameter consists of the following sub-parameters

List of APIs

This parameter contains the APIs which differ from the requested configuration.

API

This parameter contains the value for the equivalent attribute of the ASE.

List of Modules

This parameter contains the modules which differ from the requested configuration.

Slot Number

This parameter contains the value for the equivalent attribute of the ASE.

Module Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Module State

This parameter contains the value for the equivalent attribute of the ASE.

List of Submodules

This parameter list consists of the following parameter:

Subslot Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule State

This parameter contains the value for the equivalent attribute of the ASE.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter shall have the value PNIO.

Type: Unsigned8

Error code 1

The Error code 1 assumes one of the following values:

FAULTY_CONTROL_BLOCK

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 2 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 3 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.6.3.6 Prm End

This service shall be used by the client (IO controller or IO Parameter server) to inform the IO device that all parameter are written by means of the Write service of the Record Data ASE.

Special case: If the attribute AR State of the desired AR is set to BACKUP, all data records are neglected during the parameterization sequence (see 7.3.1.3.5 item b)). Thus if the AR state is set to BACKUP when the service Prm End is initiated, the Submodule State.AR Info of all submodules related to the parameterization sequence is set to the submodule's state according to the AR with AR state set to PRIMARY or if not existing set to FIRST). If submodules are added (Configuration in RUN), which are not yet contained in the PRIMARY or FIRST AR, and the submodule is existing or a decent substitute, the submodule's AR state is set to APPLICATION_READY_PENDING.

Table 181 shows the parameters of the service.

Table 181 – Prm End

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Session Key	M	M(=)		
Alarm Sequence Number	U	U(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Session Key			M	M(=)
Alarm Sequence Number			C	C(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 1 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Alarm Sequence Number

This parameter shall only be used in conjunction with a plug or pull alarm. The value shall be taken from the Alarm Notification.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter shall have the value PNIO.

Type: Unsigned8

Error code 1

The Error code 1 assumes one of the following values:
FAULTY_END_OF_PARAMETER_BLOCK

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 2 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 3 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.6.3.7 Application Ready

By this service the user of an IO device indicates to the IO controller that the parameterization is done and the input data are updated.

This service is used in the context of

- Parameterization driven by application relation establishment,
- Parameterization driven by a plug/release of a submodule, and
- Parameterization driven by a Prm Begin / Prm End sequence.

Table 182 shows the parameters of the service.

Table 182 – Application Ready

Parameter name	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Session Key	M	M(=)		
Alarm Sequence Number	U	U(=)		
Module Diff Block	U	U(=)		
List of APIs	M	M(=)		
API	M	M(=)		
List of Modules	M	M(=)		
Slot Number	M	M(=)		
Module Ident Number	M	M(=)		
Module State	M	M(=)		
List of Submodules	M	M(=)		
Subslot Number	M	M(=)		
Submodule Ident Number	M	M(=)		
Submodule State	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)

Parameter name	req	ind	rsp	cnf
Session Key			M	M(=)
Alarm Sequence Number			C	C(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

Session Key

The value of the parameter Session Key shall be increased by one for each connect by the CM Initiator. The CM Responder shall use this value for each subsequent response within this session.

NOTE 1 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Alarm Sequence Number

This parameter shall only be used in conjunction with parameterization driven by a plug/release of a submodule. The value shall be taken from the Plug/Released Alarm Notification.

Module Diff Block

The Module Diff Block contains those submodules which differ from the requested ones or have diagnosis; that are

- submodules not matching the expected configuration (wrong, missing or substituted),
- submodules not being part of the application relation (superordinated locked, locked by other IO controller or locked by supervisor),
- submodules having diagnosis, maintenance or qualified information, and
- submodules which are not available due to e.g. parameter checking pending (Application Ready Pending).
- Special case: If the attribute AR State of the desired AR is set to BACKUP, all data records are neglected during the parameterization sequence (see 7.3.1.3.5 item b)). Thus if the AR state is set to BACKUP when the service Prm End is initiated, the Submodule State.AR Info of all submodules related to the parameterization sequence is set to the submodule's state according to the AR with AR state set to PRIMARY or if not existing set to FIRST). If submodules are added (Configuration in RUN), which are not yet contained in the PRIMARY or FIRST AR, and the submodule is existing or a decent substitute, the submodule's AR state is set to APPLICATION_READY_PENDING.

An empty block is not delivered.

The IOxS of all submodules returned in the module diff block, where the Submodule State.AR Info is set to APPLICATION_READY_PENDING, SUPERORDINATED_LOCKED, LOCKED_BY_IO_CONTROLLER or LOCKED_BY_IO_SUPERVISOR, shall be set to BAD.

If such submodules are released (from AR Info equal to SUPERORDINATED_LOCKED, LOCKED_BY_IO_CONTROLLER or LOCKED_BY_IO_SUPERVISOR; indicated by a Release alarm) or the parameterization is finished (from AR Info equal to APPLICATION_READY_PENDING; indicated by a Return of Submodule alarm), the IOxS maybe set to GOOD again.

The content of the Module Diff Block depends on the context of the Application Ready:

- Parameterization driven by application relation establishment:
The Module Diff Block contains only those submodules which have been requested due to the Expected Submodule Block of the related Connect service.
- Parameterization driven by a plug/release of a submodule, and
The Module Diff Block contains the submodule which has been plugged or released due to the related Alarm Notification service.
- Parameterization driven by a Prm Begin / Prm End sequence.
The Module Diff Block contains only those submodules which have been requested due to the Expected Submodule Block of the related Prm Begin service.

If during the parameterization sequence a internal state change of the submodule takes place and invalidates the parameterization, the parameterization sequence shall be continued (with issuing the appropriate errors) and the submodule's AR Info shall be set to APPLICATION_READY_PENDING, as long as the submodule still exists and is owned by the application relation.

The parameter consists of the following sub-parameters:

List of APIs

This parameter contains a list of APIs for the modules which differ from the requested configuration. A list element consists of the following parameter.

API

This parameter contains the value for the equivalent attribute of the ASE.

List of Modules

This parameter contains a list of modules which differ from the requested configuration. A list element consists of the following parameter.

Slot Number

This parameter contains the value for the equivalent attribute of the ASE.

Module Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Module State

This parameter contains the value for the equivalent attribute of the ASE.

List of Submodules

This parameter contains a list of submodules which differ from the requested configuration. A list element consists of the following parameter.

Subslot Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule State

This parameter contains the value for the equivalent attribute of the ASE.

Result(+)

This parameter indicates that the service request succeeded.

Result(-)

This parameter indicates that the service request failed.

Error Decode

This parameter shall have the value PNIO.

Type: Unsigned8

Error code 1

The Error code 1 assumes one of the following values:

FAULTY_APPLICATION_READY_BLOCK

Type: Unsigned16

Error code 2

The parameter Error code 2 is user specific.

Type: Unsigned8

Add Data 1

The parameter Add Data is API specific (profile). The value 0 shall be transmitted if no additional data 1 is defined.

Type: Unsigned 16

NOTE 2 Add Data 1 can be used by profile specifications to transmit specific error messages.

Add Data 2

The parameter Add Data is user specific. The value 0 shall be transmitted if no additional data 2 is defined.

Type: Unsigned 16

NOTE 3 Add Data 2 can be used by device vendors to transmit specific error messages.

7.3.6.3.8 General Data Definition for Identification Data

All services to read the identification data (expected or real) have the same parameter layout. To ease the description the parameter is thus described once upfront to the services. The services itself act as a filter for the information from the context ASE.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies.

The parameter Data has a service specific structure and meaning. Table 183 shows the specific elements of Data for this service.

Table 183 – General Data definition for identification services

Parameter name	service specific elements	rsp	cnf
Data	List of APIs	M	M(=)
	API	M	M(=)
	List of Slots	M	M(=)
	Slot Number	M	M(=)
	Module Ident Number	M	M(=)
	List of Subslots	M	M(=)
	Subslot Number	M	M(=)
	Submodule Ident Number	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of APIs

This parameter is composed of the following list elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

List of Slots

This parameter is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Module Ident Number

This parameter contains the value corresponding attribute of the ASE object.

List of Subslots

This parameter is composed of the following list elements:

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Submodule Ident Number

This parameter contains the value corresponding attribute of the ASE object.

7.3.6.3.9 Read Expected Identification Data for one subslot

This confirmed service may be used to read the expected identification data (configured module and submodule identification) of a particular subslot related to a particular AR of the device. The expected identification is configured by the engineering system. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Expected Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the expected identification belonging to the addressed subslot regarding the addressed application relation.

7.3.6.3.10 Read Real Identification Data for one subslot

This confirmed service may be used to read the real identification data (actual module and submodule identification) of a particular subslot related to a particular AR of the device. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Real Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5

applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the real identification data belonging to the addressed subslot.

7.3.6.3.11 Read Expected Identification Data for one slot

This confirmed service may be used to read the expected identification data (configured module and submodule identification) of all subslots belonging to a particular slot and related to a particular AR of the device. The expected identification is configured by the engineering system. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Expected Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the expected identification belonging to the addressed slot regarding the addressed application relation.

7.3.6.3.12 Read Real Identification Data for one slot

This confirmed service may be used to read the real identification data (actual module and submodule identification) of all subslots belonging to a particular slot and related to a particular AR of the device. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Real Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the real identification data belonging to the addressed slot.

7.3.6.3.13 Read Expected Identification Data for one AR

This confirmed service may be used to read the expected identification data (configured module and submodule identification) of all subslots belonging to a particular AR of the device. The expected identification is configured by the engineering system. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Expected Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5

applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the expected identification belonging to the addressed application relation.

7.3.6.3.14 Read Real Identification Data for one AR

This confirmed service may be used to read the real identification data (actual module and submodule identification) of all subslots belonging to a particular AR of the device. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Real Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the real identification data belonging to the addressed application relation.

7.3.6.3.15 Read Real Identification Data for one API

This confirmed service may be used to read the real identification data (actual module and submodule identification) of all subslots belonging to a particular API of the device. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

Implicit AR using Target AR UUID: This service does only contain Real Identification Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is returned.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.6.3.8 for its general description.

This service filters the context object regarding the real identification data belonging to the addressed API.

7.3.6.3.16 Read Module Diff Block

This confirmed service may be used to read the actual state of the module diff block related to the expected configuration. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

By means of the service parameter Target AR UUID the content of the module diff block is restricted to a specific AR.

Implicit AR using Target AR UUID: This service does only contain a Module Diff Block if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is responded.

If no difference exists the Read Module Diff Block service shall respond no data with Length zero.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 184 shows the specific elements of Data for this service.

Table 184 – Data elements of Read Module Diff Block

Parameter name	service specific elements	rsp	cnf
Data	Module Diff Block	C	C(=)
	List of APIs	M	M(=)
	API	M	M(=)
	List of Slots	M	M(=)
	Slot Number	M	M(=)
	Module Ident Number	M	M(=)
	Module State	M	M(=)
	List of Subslots	M	M(=)
	Subslot Number	M	M(=)
	Submodule Ident Number	M	M(=)
	Submodule State	M	M(=)

Data

This parameter contains the values of the object which has been read.

Module Diff Block

The Module Diff Block contains those submodules which differ from the requested ones or have diagnosis; that are

- submodules not matching the expected configuration (wrong, missing or substituted),
- submodules not being part of the application relation (superordinated locked, locked by other IO controller or locked by supervisor),
- submodules having diagnosis, maintenance or qualified information, and
- submodules which are not available due to e.g. parameter checking pending (Application Ready Pending).

An empty block is not delivered.

The IOxS of all submodules returned in the module diff block, where the Submodule State.AR Info is set to APPLICATION_READY_PENDING, SUPERORDINATED_LOCKED, LOCKED_BY_IO_CONTROLLER or LOCKED_BY_IO_SUPERVISOR, shall be set to BAD.

If such submodules are released (from AR Info equal to SUPERORDINATED_LOCKED, LOCKED_BY_IO_CONTROLLER or LOCKED_BY_IO_SUPERVISOR; indicated by a Release alarm) or the parameterization is finished (from AR Info equal to APPLICATION_READY_PENDING; indicated by a Return of Submodule alarm), the IOxS maybe set to GOOD again.

The Module Diff Block contains only those submodules which have been requested due to the Expected Submodule Block of the related application relation indicated by Target AR UUID.

The parameter consists of the following sub-parameters:

List of APIs

This parameter contains a list of APIs for the modules which differ from the requested configuration. A list element consists of the following parameter.

API

This parameter contains the value for the equivalent attribute of the ASE.

List of Modules

This parameter contains a list of modules which differ from the requested configuration. A list element consists of the following parameter.

Slot Number

This parameter contains the value for the equivalent attribute of the ASE.

Module Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Module State

This parameter contains the value for the equivalent attribute of the ASE.

List of Submodules

This parameter contains a list of submodules which differ from the requested configuration. A list element consists of the following parameter.

Subslot Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule Ident Number

This parameter contains the value for the equivalent attribute of the ASE.

Submodule State

This parameter contains the value for the equivalent attribute of the ASE.

The following parameters have a specific meaning at this service:

Target AR UUID

This parameter shall only be used to read AR specific expected identification information. If this parameter is used the service parameter Slot Number and Subslot Number shall not be used.

NOTE It is used to read only diagnosis information the requested AR is connected to.

7.3.6.3.17 Read API Data

This confirmed service may be used to read the supported application profiles. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 185 shows the specific elements of Data for this service.

Table 185 – Data elements of Read API Data

Parameter name	service specific elements	rsp	cnf
Data	List of APs	M	M(=)
	API	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of APs

This parameter is composed of the following list elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.18 Read I&M0 Filter Data

This confirmed service may be used to read the value of the attribute I&M0 Filter Data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 186 shows the specific elements of Data for this service.

The I&M0 Filter Data indicates all submodules being a carrier of discrete identification and maintenance data. I&M1 .. I&M15 are, if supported, only writeable on such submodules. Optionally the submodules acting as module representative are indicated.

The I&M0 Filter Data returns one device representative on which I&M1, I&M2, I&M3 and I&M4 are writeable.

Table 186 – Data elements of Read I&M0 Filter Data

Parameter name	service specific elements	rsp	cnf
Data	I&M0 Filter Data	M	M(=)
	I&M0 Carrier Data	M	M(=)
	List of APs	M	M(=)
	API	M	M(=)
	List of Slots	M	M(=)
	Slot Number	M	M(=)
	Module Ident Number	M	M(=)
	List of Subslots	M	M(=)
	Subslot Number	M	M(=)
	Submodule Ident Number	M	M(=)
	Module Representative Data	O	O(=)
	List of APs	M	M(=)
	API	M	M(=)
	List of Slots	M	M(=)
	Slot Number	M	M(=)
	Module Ident Number	M	M(=)
	List of Subslots	M	M(=)
	Subslot Number	M	M(=)
	Submodule Ident Number	M	M(=)
	Device Representative Data	M	M(=)
	List of APs	M	M(=)
	API	M	M(=)
	List of Slots	M	M(=)
	Slot Number	M	M(=)
	Module Ident Number	M	M(=)
	List of Subslots	M	M(=)
	Subslot Number	M	M(=)
	Submodule Ident Number	M	M(=)

Data

This parameter contains the values of the object which has been read.

I&M0 Filter Data

This attribute consists of the following attributes:

I&M0 Carrier Data

This attribute contains all submodules being carrier of identification and maintenance data (see 7.1.4.5.7 for the definition).

It consists of the following attributes.

List of APs

This attribute is composed of the following list elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Module Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Submodule Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

Module Representative Data

This optional attribute indicates the submodules acting as module representative and thus carrying the module's I&M data; only one submodule being carrier of identification and maintenance information shall be returned per module.

It consists of the following attributes.

List of APs

This attribute is composed of the following list elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Module Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Submodule Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

Device Representative Data

This attribute contains one submodule, which is a carrier of identification and maintenance information (see 7.1.4.5.7 for the definition) and provides access to I&M1, I&M2, I&M3 and I&M4.

This attribute consists of the following attributes.

List of APs

This attribute is composed of the following list elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

List of Slots

This attribute defines the slots of the application process. One Slot is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Module Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

List of Subslots

This attribute consists of the following attributes.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Submodule Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.19 Read I&M0 Data

This confirmed service may be used to read I&M0 Data. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 187 shows the specific elements of Data for this service.

Table 187 – Data elements of Read I&M0 Data

Parameter name	service specific elements	rsp	cnf
Data	I&M0 Data	C	C(=)

Parameter name	service specific elements	rsp	cnf
	Vendor ID	M	M(=)
	Order ID	M	M(=)
	Serial Number	M	M(=)
	Hardware Revision	M	M(=)
	Software Revision	M	M(=)
	Revision Counter	M	M(=)
	Profile ID	M	M(=)
	Profile Specific Type	M	M(=)
	I&M Version	M	M(=)
	I&M Supported	M	M(=)

Data

This parameter contains the values of the object which has been read.

I&M0 Data

This parameter consists of the following elements.

Vendor ID

This parameter contains the value of the corresponding attribute of the ASE object.

Order ID

This parameter contains the value of the corresponding attribute of the ASE object.

Serial Number

This parameter contains the value of the corresponding attribute of the ASE object.

Hardware Revision

This parameter contains the value of the corresponding attribute of the ASE object.

Software Revision

This parameter contains the value of the corresponding attribute of the ASE object.

Revision Counter

This parameter contains the value of the corresponding attribute of the ASE object.

Profile ID

This parameter contains the value of the corresponding attribute of the ASE object.

Profile Specific Type

This parameter contains the value of the corresponding attribute of the ASE object.

I&M Version

This parameter contains the value of the corresponding attribute of the ASE object.

I&M Supported

This parameter contains the value of the corresponding attribute of the ASE object.

The following parameters have a specific meaning at this service:

Length (Result(+))

The parameter Length indicates the number of octets of the I&M0 Data read.

Special case (legacy submodules / modules):

The value zero, in conjunction with an entry in the I&M0 Carrier Data, indicates, that the submodule is not able to retrieve its I&M0 data.

NOTE In this case the engineering system may use locally generated I&M0 data.

7.3.6.3.20 Write I&M1 Data

This confirmed service may be used to write the attributes of the I&M1 Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 188 shows the specific elements of Data for this service.

Table 188 – Data elements of Write I&M1 Data

Parameter name	service specific elements	req	ind
Data	I&M1 Data	M	M(=)
	Function	M	M(=)
	Location	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

I&M1 Data

This parameter consists of the following elements.

Function

This parameter contains the value of the corresponding attribute of the ASE object.

Location

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.21 Read I&M1 Data

This confirmed service may be used to read I&M1 Data. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 189 shows the specific elements of Data for this service.

Table 189 – Data elements of Read I&M1 Data

Parameter name	service specific elements	rsp	cnf
Data	I&M1 Data	C	C(=)
	Function	M	M(=)
	Location	M	M(=)

Data

This parameter contains the values of the object which has been read.

I&M1 Data

This parameter consists of the following elements.

Function

This parameter contains the value of the corresponding attribute of the ASE object.

Location

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.22 Write I&M2 Data

This confirmed service may be used to write the attributes of the I&M2 Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 190 shows the specific elements of Data for this service.

Table 190 – Data elements of Write I&M2 Data

Parameter name	service specific elements	req	ind
Data	I&M2 Data	M	M(=)
	Installation Date	M	M(=)

Data

This parameter contains the value of the object which may be used to write

I&M2 Data

This parameter consists of the following elements

Installation Date

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.23 Read I&M2 Data

This confirmed service may be used to read I&M2 Data. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 191 shows the specific elements of Data for this service.

Table 191 – Data elements of Read I&M2 Data

Parameter name	service specific elements	rsp	cnf
Data	I&M2 Data	C	C(=)
	Installation Date	M	M(=)

Data

This parameter contains the value of the object which has been read.

I&M2 Data

This parameter consists of the following elements.

Installation Date

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.24 Write I&M3 Data

This confirmed service may be used to write the attributes of the I&M3 Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 192 shows the specific elements of Data for this service.

Table 192 – Data elements of Write I&M3 Data

Parameter name	service specific elements	req	ind
Data	I&M3 Data	M	M(=)
	Descriptor	M	M(=)

Data

This parameter contains the value of the object which may be used to write.

I&M3 Data

This parameter consists of the following elements.

Descriptor

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.25 Read I&M3 Data

This confirmed service may be used to read I&M3 Data. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 193 shows the specific elements of Data for this service.

Table 193 – Data elements of Read I&M3 Data

Parameter name	service specific elements	rsp	cnf
Data	I&M3 Data	C	C(=)
	Descriptor	M	M(=)

Data

This parameter contains the value of the object which has been read.

I&M3 Data

This parameter consists of the following elements.

Descriptor

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.26 Write I&M4 Data

This confirmed service may be used to write the attributes of the I&M4 Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 194 shows the specific elements of Data for this service.

If the I&M4 Data is used for profile specific purposes (e.g. PROFIsafe), the data may only be written by the IO device internally upon manufacturer defined criteria (safety concepts) after fulfilled parameterization. In such case the Write I&M4 Data service is denied.

Table 194 – Data elements of Write I&M4 Data

Parameter name	service specific elements	req	ind
Data	I&M4 Data	M	M(=)

Parameter name	service specific elements	req	ind
	Signature	M	M(=)

Data

This parameter contains the value of the object which may be used to write.

I&M4 Data

This parameter consists of the following elements.

Signature

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.27 Read I&M4 Data

This confirmed service may be used to read I&M4 Data. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 195 shows the specific elements of Data for this service.

Table 195 – Data elements of Read I&M4 Data

Parameter name	service specific elements	rsp	cnf
Data	I&M4 Data	C	C(=)
	Signature	M	M(=)

Data

This parameter contains the value of the object which has been read.

I&M4 Data

This parameter consists of the following elements.

Signature

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.28 Read Autoconfiguration Data

This confirmed service may be used to read the autoconfiguration data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 196 shows the specific elements of Data for this service.

If no Autoconfiguration Data exists the Read Autoconfiguration Data service shall respond no data with length zero.

Table 196 – Data elements of Read Autoconfiguration Data

Parameter name	service specific elements	rsp	cnf
Data	Autoconfiguration Data	M	M(=)
	Communication Data	M	M(=)
	Min Device Interval	M	M(=)
	Communication Properties	M	M(=)

Parameter name	service specific elements	rsp	cnf
	DFP	M	M(=)
	RTC3	M	M(=)
	Send Clock Factors	M	M(=)
	Communication Data	M	M(=)
	List of Offered Submodules	M	M(=)
	API	M	M(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	Module Ident Number	M	M(=)
	Submodule Ident Number	M	M(=)
	Length Input Data	M	M(=)
	Length Output Data	M	M(=)
	Isochronous Data	C	C(=)
	List of Offered Submodules	M	M(=)
	API	M	M(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	Time IO Base	M	M(=)
	Time IO Input	M	M(=)
	Time IO Output	M	M(=)

Data

This parameter contains the values of the object which has been read.

Autoconfiguration Data

This parameter consists of the following elements:

Communication Data

This parameter consists of the following elements:

Min Device Interval

This parameter contains the value of the corresponding attribute of the ASE object.

Communication Properties

This parameter contains the supported communication properties.

DFP

This parameter contains the value whether DFP is supported or not.

RTC3

This parameter contains the value whether PDIRFrameData with local and/or forwarding information is supported

RTCUUDP

This parameter contains the value whether RT_CLASS_UDP is supported or not.

Send Clock Factors

This parameter contains the value of the corresponding attribute of the ASE object.

Configuration Data

This parameter consists of the following elements:

List of Offered Submodules

This list consists of the following elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Module Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

Submodule Ident Number

This parameter contains the value of the corresponding attribute of the ASE object.

Length Input Data

This parameter contains the value of the corresponding attribute of the ASE object.

Length Output Data

This parameter contains the value of the corresponding attribute of the ASE object.

Configuration Data

This parameter consists of the following elements:

List of Offered Submodules

This list consists of the following elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Base

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Input

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Output

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.6.3.29 Write Expected Fast Startup Data

This confirmed service may be used to write the expected fast startup data of the application relation. This service shall be used in conjunction with the IO AR. This service is only

supported for ARs in legacy startup mode; the service is substituted for ARs in advanced startup mode by the AR FSU Block within the connect service.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 197 shows the specific elements of Data for this service.

The service user shall only address modules and submodules within the port data, which are part of the submodule list the AR is connected to. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored. However, the service user may address fewer submodules. In this case the port difference service provides the real data of the missing ports.

Table 197 – Data elements of Write Expected Fast Startup Data

Parameter name	service specific elements	req	ind
Data	Expected Fast Startup Data	M	M(=)
	FS Parameter Block	O	O(=)
	FS Parameter Mode	M	M(=)
	FS Parameter UUID	M	M(=)
	Fast Startup Block	O	O(=)
	Data	M	M(=)

Data

This parameter contains the value of the object which may be used to write.

Expected Fast Startup Data

This parameter consists of the following elements:

FS Parameter Block

This optional parameter consists of the following elements:

FS Parameter Mode

This parameter consists of the following elements:

Mode

Mode is used to establish the fast startup mode of the application.

Attribute type: Unsigned8

Allowed values: OFF (0x00), ON (0x01)

Default value: OFF

FS Parameter UUID

This parameter contains a UUID indicating the generation of the record data configuration for the device. The UUID is changed by engineering whenever something within the record data changes, records are added or subtracted from the configuration.

Attribute type: UUID

Fast Startup Block

This optional parameter consists of the following elements:

Data

This parameter holds data.

Attribute type: OctetString

7.3.6.3.30 Read Expected Fast Startup Data

This confirmed service may be used to read the expected fast startup data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR. This service is only supported for ARs in legacy startup mode; the service is substituted for ARs in advanced startup mode by the AR FSU Block within the connect service.

By means of the service parameter Target AR UUID the content of the port difference can be restricted to a specific AR as a filter function using the Implicit AR. In case a user specific filter is selected the Read FSU Data response shall only contain items that match the filter criteria. Otherwise all items shall be responded.

Implicit AR using Target AR UUID: This service does only contain FSU Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is responded.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 198 shows the specific elements of Data for this service.

Table 198 – Data elements of Read Expected Fast Startup Data

Parameter name	service specific elements	rsp	cnf
Data	Expected Fast Startup Data	M	M(=)
	FS Parameter Block	C	C(=)
	FS Parameter Mode	M	M(=)
	FS Parameter UUID	M	M(=)
	Fast Startup Block	C	C(=)
	Data	M	M(=)

Data

This parameter contains the value of the object which has been read.

Expected Fast Startup Data

This parameter consists of the following elements:

FS Parameter Block

This optional parameter consists of the following elements:

FS Parameter Mode

This parameter consists of the following elements:

Mode

Mode is used to establish the fast startup mode of the application.

Attribute type: Unsigned8

Allowed values: OFF (0x00), ON (0x01)

Default value: OFF

FS Parameter UUID

This parameter contains a UUID indicating the generation of the record data configuration for the device. The UUID is changed by engineering whenever something within the record data changes, records are added or subtracted from the configuration.

Attribute type: UUID

Fast Startup Block

This optional parameter consists of the following elements:

Data

This parameter holds data.

Attribute type: OctetString

7.3.6.3.31 Read GSD Data

This confirmed service may be used to read the GSD data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read query service (see 7.3.1.4.1.2); for the error code hierarchy of the record data object 7.3.1.3.5 applies. The parameter Data has a service specific structure and meaning; see 7.3.4.3.4 for its general description.

GSD data belongs to the device. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

The model to upload the GSD is described as follows:

- The uploader starts with From Offset Data set to "0" (i.e. start uploading at the begin) and specifies in the Read Query request the maximum length to read in one chunk via the parameter Length
- The IO device responds with Data set according to Table 199; mirrors the requested From Offset Data, specifies for the uploader which value to use for the next call to this service as Next Offset Data, and informs the uploader about the Total Size. The Data returned is a chunk from the ASE's attribute GSD Data (starting with GSD Data[From Offset Data] up to GSD Data[Next Offset Data]).
- If Next Offset Data is returned with a value "0", the last chunk of the GSD Data has been read

Table 199 shows the specific elements of Selector for this service.

Table 199 – Data elements of Read GSD Data

Parameter name	service specific elements	rsp	cnf
Selector	From Offset Data	M	M(=)

Data

This parameter contains the value of the object which has been read.

From Offset Data

Specifies the start offset for the GSD data.

Attribute type: Unsigned32

Table 200 shows the specific elements of Data for this service.

Table 200 – Data elements of Read GSD Data

Parameter name	service specific elements	rsp	cnf
Data	From Offset Data	M	M(=)
	Next Offset Data	M	M(=)
	Total Size	M	M(=)
	Data	M	M(=)

Data

This parameter contains the value of the object which has been read.

From Offset Data

Mirrors the From Offset Data of the request.

Attribute type: Unsigned32

Next Offset Data

Specifies the next offset to read the GSD data.

Attribute type: Unsigned32

Special case: 0, if last chunk of GSD data has been transmitted

Total Size

Specifies the total size of the GSD data.

Attribute type: Unsigned32

Data

Chunk of GSD Data.

7.3.6.4 Behavior of the Context object

The Context object contains the real identification data from the perspective of the IO device. The real identification is detected by local means before the IO device may communicate. This real identification may be changed during operation by a local means. It may affect adding or removing of slots with their subslots. In such a case, the real identification shall be updated and the IO controllers or IO supervisor connected via affected ARs shall be informed by means of an Alarm service. The Alarm Types “Released”, “Controlled By Supervisor”, “Pull Alarm”, “Plug Alarm” or “Plug Wrong Submodule” shall be used to indicate such events.

An affected AR means an established relationship where at least one of the changed slot/subslot is part of the attribute Expected identification.

When receiving the Alarm Notification with Alarm type “Plug Alarm” the IO controller shall check if the new added slot/subslot needs additional parameter and convey this parameter to the IO device. The IO controller may also inform the local user.

When receiving the Alarm Notification with Alarm type “Pull Alarm” or “Plug Wrong Submodule” the IO controller may inform the local user and wait for further actions.

However, in the time between a Connect response and receiving the Prm End indication, the real identification shall be kept consistent in any case. If “real” modules are removed within this time interval then it shall be reported afterwards by means of Application Ready. Furthermore, if “real” modules are plugged within this time interval then it shall be reported afterwards by means of Alarm Notifications. It is not allowed to respond inconsistent or not plausible Module Diff Blocks with the Connect response or Application Ready request service primitives.

NOTE The following example illustrates an inconsistent Module Diff Block. If a submodule state was reported in the Connect response it could not be missed in the following Application Ready request, even if it was plugged in between. However, a submodule can change to a bad state because of faulty submodule parameter.

7.3.6.5 Behavior of the attribute Submodule State

The attribute behavior of an IO device application is described in 7.4.6.

7.3.7 Isochronous Mode Application ASE

7.3.7.1 Overview

This ASE specifies the structure of the necessary parameters for isochronous mode of applications. The IsoM ASE provides a set of services used to read, and write their values. This ASE is an option and does not effect non-isochronous applications.

Figure 33 shows the basic model for isochronous applications:

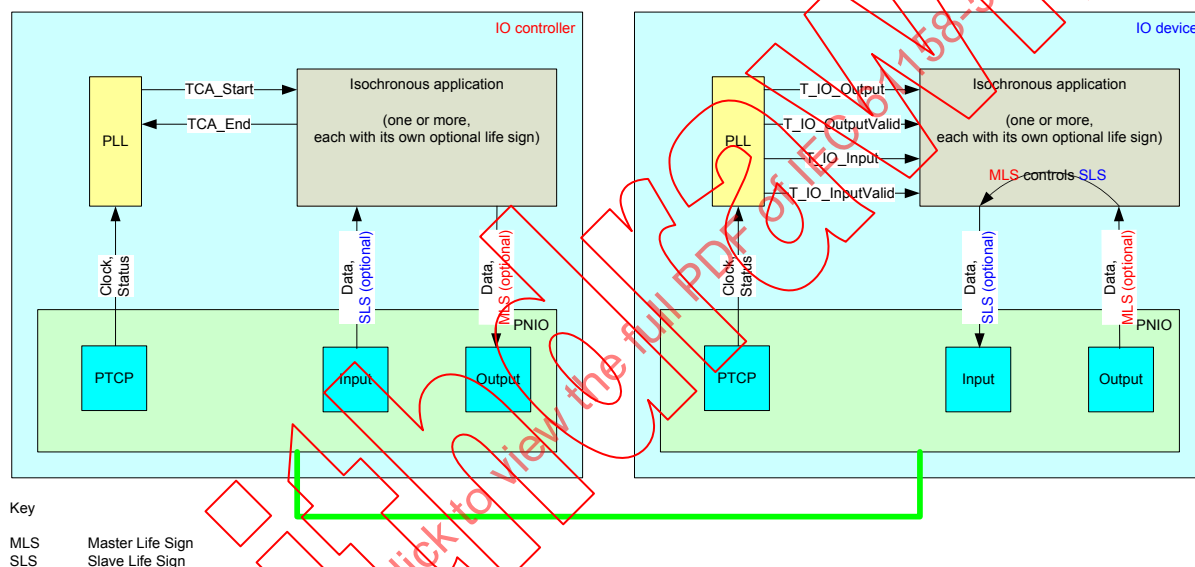


Figure 33 – Basic model for isochronous applications

Isochronous Mode couples signal measurement, transfer and processing, transfer and signal output to the conveyance of the fieldbus. Isochronous mode is application coupling and not an attribute of the transmitting media.

Thus one should **not** confuse isochronous mode with an equidistant fieldbus. In fact it is possible to model isochronous mode onto a non-equidistant fieldbus or run applications freely on top of an equidistant fieldbus. However to reduce complexity isochronous mode is modeled typically onto an equidistant fieldbus, to make the transfer of data robust and deterministic.

Isochronous mode means, that a bunch of peripheral inputs, their processing within an application program on the controller and a bunch of peripheral outputs are related to a common pulse.

Figure 34 shows the means for isochronous mode; in particular the common pulse is defined by the start of the data cycle (T_DC), T_IO_Input relates peripheral inputs and T_IO_Output relates peripheral outputs to the data cycle. The isochronous application within the IO controller is related to the data cycle by local means.

For increasing the transmission's robustness data loss is accepted even within an equidistant fieldbus. The data hold time (as configured by the user) defines the maximum number of data

Thus a data actuality window is opened. The application per se does not know, whether a specific peripheral input value is related to this current data cycle or to some past data cycle. This applies also to a specific peripheral output; again the output does not know, whether the data value to be set is out of this current data cycle or from some past data cycle. The actuality of data may be spread over 2 data cycles without getting any notice of the peripheral bus system.

If on base of a data actuality window an isochronous application is to be modeled, the application needs means to uncover whether data is from the current cycle or some past cycle. For this the application cycles are synchronized by application means: the so called **“Life Sign”** (Master Life Sign and Slave Life Sign).

Figure 34 shows the relation of the ASE attributes within the isochronous application model.

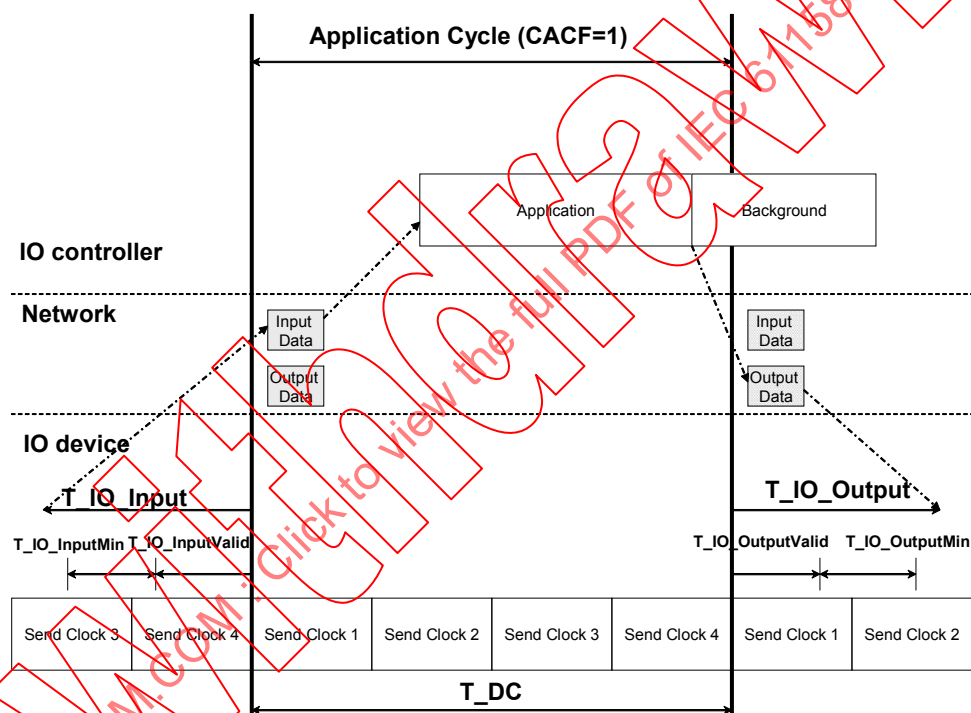


Figure 34 – General isochronous application model (example CACF == 1)

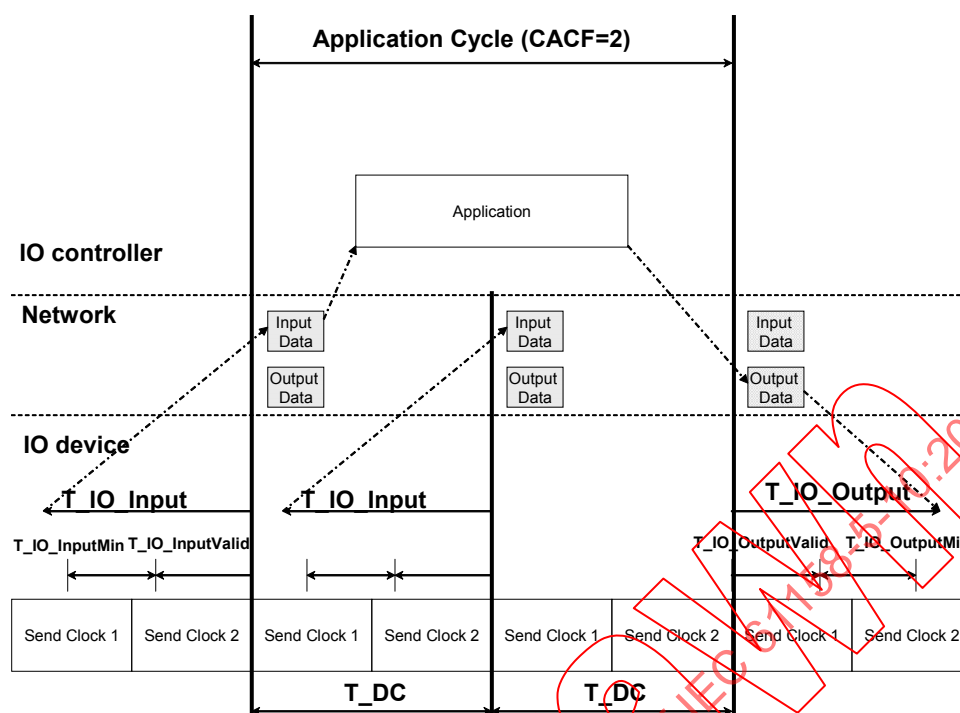


Figure 35 – General isochronous application model (example CACF == 2)

The application in the IO controller is started with a reduction to the data cycle (CACF). It may thus run in parallel to the data cycle (if CACF == 1) or reduced to the data cycle (if CACF > 1). The controller application provides new output values every controller application cycle ($CACF * T_{DC} * T_{DC\_BASE} * 31,25 \mu s$).

The data from/for a particular submodule is transported via RTC data services in every data cycle (T_{DC}). The application in the IO device runs in every data cycle; thus in every data cycle new input values are available for the IO controller ($T_{DC} * T_{DC\_BASE} * 31,25 \mu s$).

New output values are optionally indicated by the IO controller by a change of the MLS (master life sign) which is a part of the output data. The MLS changes thus in every controller application cycle ($CACF * T_{DC} * T_{DC\_BASE} * 31,25 \mu s$). The values are sampled by the IO device at T_{IO_Output} after the data cycle begin, when the new data was being received.

New input values are optionally indicated by the IO device by a change of the SLS (slave life sign) which is a part of the input data. The SLS changes thus in every data cycle ($T_{DC} * T_{DC\_BASE} * 31,25 \mu s$). The values are sampled by the IO device at T_{IO_Input} before the next data cycle begin.

7.3.7.2 Isochronous Mode Application class specification

7.3.7.2.1 Template

An Isochronous Mode Application process object is described by the following template:

ASE: Isochronous Mode Application ASE
CLASS: Isochronous Mode Application
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Key Attribute: Implicit
 2 (m) Attribute: List of Modules

2.1	(m) Attribute:	Slot Number
2.2	(m) Attribute:	List of Submodules
2.2.1	(m) Attribute:	Subslot Number
2.2.2	(m) Attribute:	IsoM Data
2.2.2.1	(m) Attribute:	Time Data Cycle
2.2.2.2	(m) Attribute:	Time IO Input
2.2.2.3	(m) Attribute:	Time IO Output
2.2.2.4	(m) Attribute:	Time IO Input Valid
2.2.2.5	(m) Attribute:	Time IO Output Valid
2.2.2.6	(m) Attribute:	Controller Application Cycle Factor
2.2.3	(m) Attribute:	Device Parameter
2.2.3.1	(m) Attribute:	Time Data Cycle Base
2.2.3.2	(m) Attribute:	Time IO Base
2.2.3.3	(m) Attribute:	Time Data Cycle Min
2.2.3.4	(m) Attribute:	Time Data Cycle Max
2.2.3.5	(m) Attribute:	Time IO Input Min
2.2.3.6	(m) Attribute:	Time IO Output Min

SERVICES:

1	(m) OpsService:	Write Isochronous Mode Data
2	(m) OpsService:	Read Isochronous Mode Data
3	(m) OpsService:	Local SYNCH Event

7.3.7.2.2 Attributes

Implicit

The key attribute Implicit indicates that the Isochronous Mode Application process object is implicitly addressed by the service.

List of Modules

This attribute list contains the following attributes:

Slot Number

This attribute defines to which module the Isochronous Mode Application attribute belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

List of Submodules

This attribute list contains the following attributes:

Subslot Number

This attribute defines to which subslot the Isochronous Mode Application attribute belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Isochronous Mode Data

This attribute consists of the following attributes:

Time Data Cycle (T_DC)

This attribute indicates the time factor for the data cycle. This time includes all parts of the isochronous data cycle to convey inputs and outputs.

The time of the data cycle is calculated by $T\_DC * T\_DC\_BASE * 31,25 \mu s$.

Time base: $T\_DC\_BASE * 31,25 \mu s$

Attribute type: Unsigned16

Allowed values: 1 to 1 024

Time IO Input (T_IO_Input)

This attribute indicates the time factor to fetch the inputs for the IO device. The time base is value of T_IO_Base multiplied with 1 ns.

The time (T_IO_Input) is related to the start of T_DC and describes a time **before** the start of T_DC.

T_IO_Input has a step size of T_IO_Base; i.e. $T\_IO\_Input = n * T\_IO\_BASE$.

For T_IO_Input the following relation shall be applied:

$$(T\_IO\_InputValid + T\_IO\_InputMin) \leq T\_IO\_Input \leq T\_DC$$

Time base: 1 ns

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time IO Output (T_IO_Output)

This attribute indicates the time factor to set the outputs for the IO device. The time base is value of T_IO_Base multiplied with 1 ns.

The time (T_IO_Output) is related to the start of T_DC and describes a time **after** the start of T_DC.

T_IO_Output has a step size of T_IO_Base, i.e. $T\_IO\_Output = n * T\_IO\_BASE$.

For T_IO_Output the following relation shall be applied:

$$(T\_IO\_OutputValid + T\_IO\_OutputMin) \leq T\_IO\_Output \leq T\_DC$$

Time base: 1 ns

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time IO Input Valid (T_IO_InputValid)

This attribute indicates the point of time the new input data shall be available to convey.

This point of time is related to the start of T_DC and describes a time **before** the start of T_DC.

T_IO_InputValid has a step size of T_IO_Base; i.e. $T\_IO\_InputValid = n * T\_IO\_BASE$.

Time base: 1 ns

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time IO Output Valid (T_IO_OutputValid)

This attribute indicates the point of time the new output data are available to process.

This point of time is related to the start of T_DC and describes a time **after** the start of T_DC.

T_IO_OutputValid has a step size of T_IO_Base; i.e. $T\_IO\_OutputValid = n * T\_IO\_BASE$.

Time base: 1 ns

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Controller Application Cycle Factor (CACF)

This attribute indicates the IO Controller Application cycle time as a factor in multiples of T_DC. It determines the time the application of the IO Controller needs to process an isochronous application task completely. From isochronous point of view there is exactly one user task for each AR controlling the isochronous part of the application.

The isochronous application task of the IO controller is started with a local time offset (the so called T_CA_START) from the cycle begin, where global cycle counter modulo $(T\_DC * CACF) == 0$.

Time base: T_DC

Attribute type: Unsigned16

Allowed values: 1 to 1024

Device Parameter

This attribute consists of the following attributes.

Time Data Cycle Base (T_DC_Base)

This attribute determines the granularity for data cycle adjustment.

This field is part of the GSD.

Time base: 31,25 μ s

Attribute type: Unsigned16

Allowed values: 1 to 1 024

Time IO Base (T_IO_Base)

This attribute determines the granularity for input and output delay time adjustment.

The time for T_IO_Base is calculated by $T\_IO\_Base * 1$ ns.

This field is part of the GSD.

Time base: 1 ns

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time Data Cycle Min (T_DC_Min)

This attribute determines the smallest possible data cycle.

The time for T_DC_Min is calculated by $T\_DC\_Min * T\_DC\_Base * 31,25$ μ s.

This field is part of the GSD.

Time base: $T\_DC\_Base * 31,25$ μ s

Attribute type: Unsigned16

Allowed values: 1 to 1 024

Time Data Cycle Max (T_DC_Max)

This attribute determines the longest possible data cycle.

The time for T_DC_Max is calculated by $T\_DC\_Max * T\_DC\_Base * 31,25 \mu s$.

This field is part of the GSD.

Time base: $T\_DC\_Base * 31,25 \mu s$

Attribute type: Unsigned16

Allowed values: 1 to 1 024

Time IO Input Min (T_IO_InputMin)

This attribute contains the factor for the smallest possible input refresh delay time. It may be the smallest possible time to copy or preprocess the input data.

This point of time is related to the start of T_DC and describes a time **before** the start of T_DC.

The time for T_IO_InputMin is calculated by $T\_IO\_InputMin * T\_IO\_Base * 1 ns$.

This field is part of the GSD.

Time base: $T\_IO\_Base * 1 ns$

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

Time IO Output Min (T_IO_OutputMin)

This attribute contains the factor for the smallest possible output refresh delay time. It may be the smallest possible time to copy or preprocess the output data.

This point of time is related to the start of T_DC and describes a time **after** the start of T_DC.

The time for T_IO_OutputMin is calculated by $T\_IO\_OutputMin * T\_IO\_Base * 1 ns$.

This field is part of the GSD.

Time base: $T\_IO\_Base * 1 ns$

Attribute type: Unsigned32

Allowed values: 1 to 32 000 000

7.3.7.3 Isochronous Mode Application service specification**7.3.7.3.1 Write Isochronous Mode Data**

This confirmed service may be used to write the attributes of the Isochronous Mode Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). But the parameter Data has a service specific structure and meaning. Table 201 shows the specific elements of Data for this service.

The service user shall only address modules and submodules within the List of Modules, which are part of the submodule list the AR is connected to. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 201 – Data elements of Write Isochronous Mode Data

Parameter name	service specific elements	req	ind
Data	Isochronous Mode Data	M	M(=)
	Time Data Cycle	M	M(=)
	Time IO Input	M	M(=)
	Time IO Output	M	M(=)
	Time IO Input Valid	M	M(=)
	Time IO Output Valid	M	M(=)
	Controller Application Cycle Factor	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Isochronous Mode Data

This parameter is composed of the following elements:

Time Data Cycle

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Input

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Output

This parameter contains the value corresponding attribute of the ASE object.

Time IO Input Valid

This parameter contains the value corresponding attribute of the ASE object.

Time IO Output Valid

This parameter contains the value corresponding attribute of the ASE object.

Controller Application Cycle Factor

This parameter contains the value corresponding attribute of the ASE object.

7.3.7.3.2 Read Isochronous Mode Data

This confirmed service may be used to read the values of the attributes of the Isochronous Mode Data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 202 shows the specific elements of Data for this service.

Implicit AR using Target AR UUID: This service does only contain Real Port Data if there is an established IO or Supervisor AR with the requested Target AR UUID. Otherwise a parameter error is responded.

Table 202 – Data elements of Read Isochronous Mode Data

Parameter name	service specific elements	rsp	cnf
Data	Isochronous Mode Data	M	M(=)
	Time Data Cycle	M	M(=)
	Time IO Input	M	M(=)
	Time IO Output	M	M(=)

Parameter name	service specific elements	rsp	cnf
	Time IO Input Valid	M	M(=)
	Time IO Output Valid	M	M(=)
	Controller Application Cycle Factor	M	M(=)

Data

This parameter contains the values of the object which has been read.

Isochronous Mode Data

This parameter is composed of the following elements:

Time Data Cycle

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Input

This parameter contains the value of the corresponding attribute of the ASE object.

Time IO Output

This parameter contains the value corresponding attribute of the ASE object.

Time IO Input Valid

This parameter contains the value corresponding attribute of the ASE object.

Time IO Output Valid

This parameter contains the value corresponding attribute of the ASE object.

Controller Application Cycle Factor

This parameter contains the value corresponding attribute of the ASE object.

The following parameters have a specific meaning at this service:

Target AR UUID

This parameter shall only be used to read AR specific port difference information. If this parameter is used the service parameter Subslot Number shall not be used.

NOTE It is used to read only port difference information the requested AR is connected to.

7.3.7.3.3 Local SYNCH Event

The service SYNCH indicates to the application of the IO device that a new isochronous send clock cycle has been started.

Table 203 shows the parameters of the service.

Table 203 – Local SYNCH Event

Parameter name	ind
Argument	M
Slot	M
Subslot	M
Global Cycle Counter	M
Phase	M
Status	M

Argument

The argument shall convey the service specific parameters of the service indication.

Slot Number

This parameter contains the Slot Number addressing the initiators interface.

Subslot Number

This parameter contains the Subslot Number addressing the initiators interface.

Global Cycle Counter

This parameter shall contain the local value of the cycle counter.

Phase

This parameter contains the Phase to be used for the RTC frame.

Status

This parameter shall contain the current status. The allowed values are LOCAL and REMOTE. The value LOCAL shall be used to indicate that no valid synchronization frame has been received within the PLL window. The value REMOTE shall be used to indicate that a valid synchronization frame has been received within the PLL window.

7.3.7.4 Behavior of Isochronous Mode Application process objects

7.3.7.4.1 Overview of relations to other ASE objects

For performance reasons, the value for the Output Data object shall be transferred on receipt as fast as possible to the object. The speed and the jitters of this action is a performance parameter of an IO device. This parameter determines the synchronization accuracy.

The Isochronous Mode Application model is a superset of the model described in IEC 61158-5-3.

Figure 36 shows the timing relations between different involved ASEs for a particular submodule.

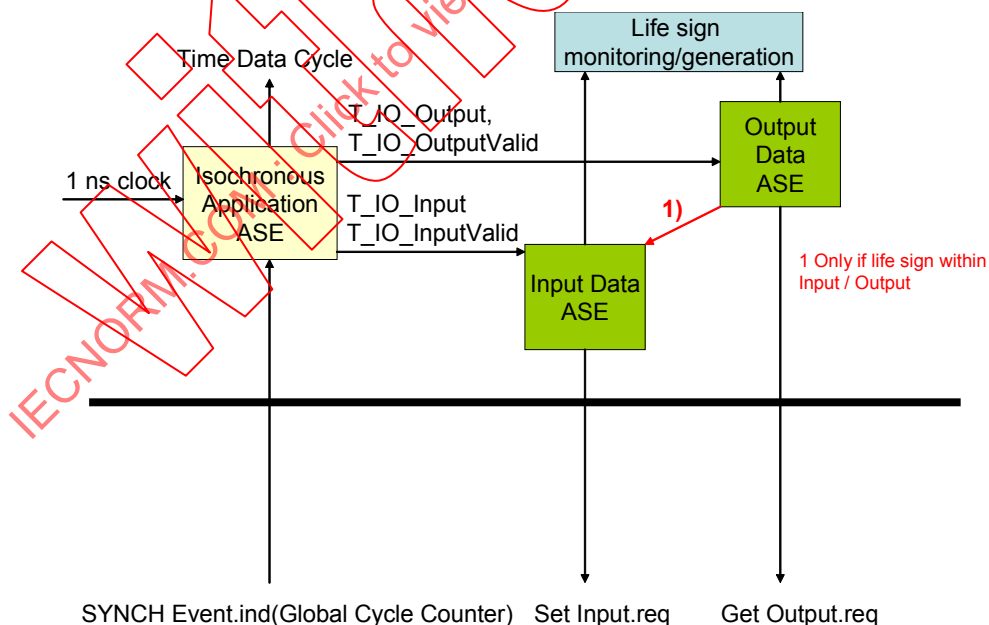


Figure 36 – ASE relations in an IO device operating in isochronous mode for a submodule

7.3.7.4.2 Basic behavior of input and output data in the isochronous mode

An IO system operating in the isochronous mode provides the following features:

- all IO devices taking part in the isochronous mode functionality synchronize their internal clock system with the synchronization message SYNCH Event with the isochronous data cycle;
- all IO devices taking part in the isochronous mode functionality have to be in the same PTCIP subdomain and furthermore in the same isochronous real-time domain;
- the jitter of SYNCH Event.ind is less than 1 μ s (jitters within an IO device is assumed to be zero).

In an IO system with enabled isochronous mode functionality IO devices with basic functionality can be used in conjunction with synchronized IO devices:

- if the SYNCH-Event.ind is issued with the status “REMOTE” then the application is synchronized with other devices;
- if the SYNCH-Event.ind is issued with the status “LOCAL” then the isochronous application shall be informed that the application may be not synchronized with other devices

7.3.7.4.3 State machine description for isochronous mode

7.3.7.4.3.1 General

The behavior of an Output Data object in case of Isochronous Mode is defined by the Isochronous Mode Output Data state machine (ISOM_OUT) state machine.

The ISOM_OUT is triggered by the Isochronous Mode Sync state machine (ISOM_SYNC).

The ISOM_OUT and ISOM_SYNC state machine represents a part of the User functionality and are included to ensure interoperability.

The ISOM_OUT and ISOM_SYNC state machine provides events to other parts of the User functionality. The ISOM_SYNC state machine has interfaces to the Isochronous Mode Input Data state machine (ISOM_IN) and the ISOM_OUT state machine for co-ordination purposes.

The main functionalities of the ISOM_OUT state machine are:

- Check the correct sequence: SYNCH, New Output, read out the Output Data at Time IO Output.
- Transfer the received Output Data to the application on time.
- Check the Controllers Sign of Life (C-LS).

The main functionalities of the ISOM_SYNC state machine are:

- Generate timing signals from the SYNCH-Event.ind
- Generation of a pulse scheme for the local application

The synchronization primitive (SYNCH Event) is conveyed by a local PLL.

7.3.7.4.3.2 ISOM_SYNC

7.3.7.4.3.2.1 Primitive definitions

7.3.7.4.3.2.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by ISOM_SYNC state machine are described in the service definition and shown in Table 204.

Table 204 – Remote primitives issued or received by ISOM_SYNC

Primitive	Source	Destination	Associated parameters	Functions
Local SYNCH Event.ind	FAL	ISOM_SYNC	see 7.3.7.3.3	—

7.3.7.4.3.2.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by ISOM_SYNC state machine are described in the service definition and shown in Table 205.

Table 205 – Local primitives issued or received by ISOM_SYNC

Primitive	Source	Destination	Associated parameters	Functions
SYNCH_Status_ind	ISOM_SYNC	ISOM_IN, ISOM_OUT	state	—
Start_O_ind	ISOM_SYNC	ISOM_OUT	—	—
SYNCH_I_ind	ISOM_SYNC	ISOM_IN	—	—
SYNCH_O_ind	ISOM_SYNC	ISOM_OUT	—	—

7.3.7.4.3.2.2 State transition diagram

Figure 37 shows the state diagram of the ISOM_SYNC state machine.

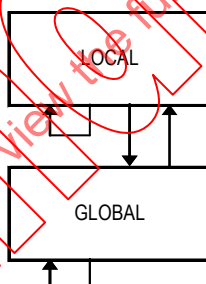


Figure 37 – State transition diagram of ISOM_SYNC

States of ISOM_SYNC

LOCAL

The clock is driven by the local time system

GLOBAL

The clock is driven by the global time system, synchronized for the corresponding devices

7.3.7.4.3.2.3 State machine description

This state machine indicates the ISOM_IN and ISOM_OUT state machines the state of the SYNCH signal (LOCAL or REMOTE) and starts the ISOM_IN and ISOM_OUT state machines.

7.3.7.4.3.2.4 State table

Table 206 shows the states of an ISOM_SYNC state machine.

Table 206 – State table ISOM_SYNC

#	Current State	Event /Condition =>Action	Next State
1	LOCAL	Local SYNCH Event.ind () /Status == LOCAL => SYNCH_Status_ind (LOCAL)	LOCAL
2	LOCAL	Local SYNCH Event.ind () /Status == GLOBAL && (GlobalCycleCounter mod SendClock*ReductionRatio) != 0 => ignore	GLOBAL
3	LOCAL	Local SYNCH Event.ind () /Status == GLOBAL && (GlobalCycleCounter mod SendClock*ReductionRatio) == 0 => Start_I_ind Start_O_ind SYNCH_I_ind SYNCH_O_ind SYNCH_Status_ind (GLOBAL)	GLOBAL
4	GLOBAL	Local SYNCH Event.ind () /Status == LOCAL => ignore	LOCAL
5	GLOBAL	Local SYNCH Event.ind () /Status == GLOBAL && (GlobalCycleCounter mod SendClock*ReductionRatio) != 0 => ignore	GLOBAL
6	GLOBAL	Local SYNCH Event.ind () /Status == GLOBAL && (GlobalCycleCounter mod SendClock*ReductionRatio) == 0 => Start_I_ind Start_O_ind SYNCH_I_ind SYNCH_O_ind SYNCH_Status_ind (GLOBAL)	GLOBAL

7.3.7.4.3.2.5 Functions, Macros, Timers and Variables

Table 207 contains the functions, macros, timers and variables used by ISOM_SYNC state machine and their arguments and their descriptions.

Table 207 – Functions, Macros, Timers and Variables used by the ISOM_SYNC

Name	Type	Function/Meaning
—	—	—

7.3.7.4.3.3 ISOM_OUT

7.3.7.4.3.3.1 Primitive definitions

7.3.7.4.3.3.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by ISOM_OUT machine are described in the service definition and shown in Table 208.

Table 208 – Remote primitives issued or received for ISOM_OUT

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	ISOM_OUT	FAL	see 7.3.5.3	—

7.3.7.4.3.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by ISOM_OUT state machine are described in the service definition and shown in Table 209.

Table 209 – Local primitives issued or received for ISOM_OUT

Primitive	Source	Destination	Associated parameters	Functions
Local Get Output_cnf	FAL	ISOM_OUT	see 7.3.2.5.12	Confirms Get Output request and passes new output data to output state machine
Local New Output_ind	FAL	ISOM_OUT	see 7.3.2.5.14	Indicates new output data has been received
Local Get Output_req	ISOM_OUT	FAL	see 7.3.2.5.12	
Start_O.ind	ISOM_SYNC	ISOM_OUT	—	Start Output Data processing
Stop_O.ind	ISOM_SYNC	ISOM_OUT	—	Stop Output Data processing
SYNCH_O.ind	ISOM_SYNC	ISOM_OUT	—	Trigger Signal for Output Data Processing

7.3.7.4.3.3.2 State transition diagram

Figure 38 shows the state diagram of the ISOM_OUT state machine.

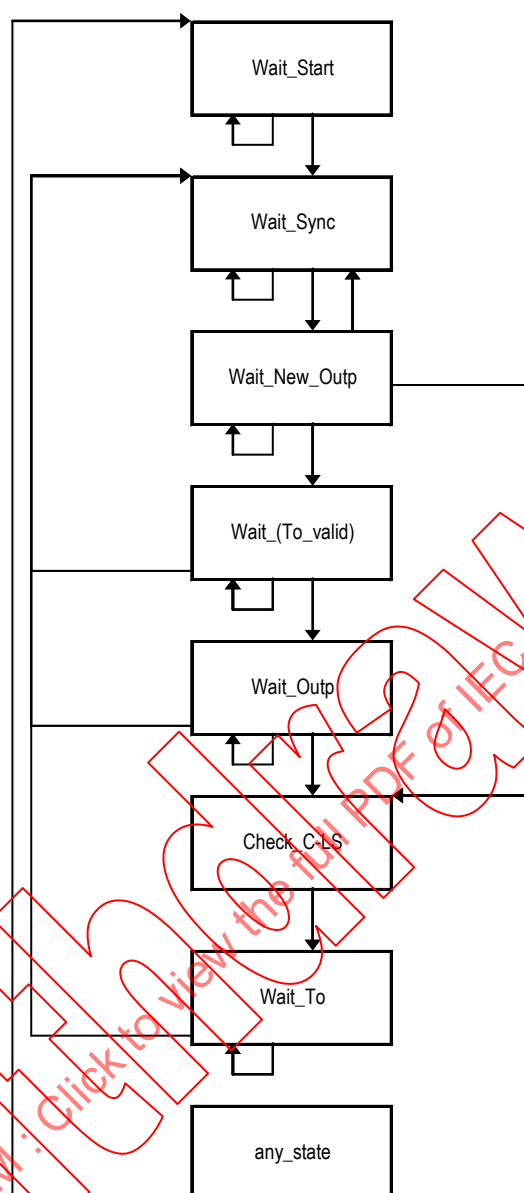


Figure 38 – State transition diagram ISOM_OUT

States of the ISOM_OUT state machine

Wait_Start	Waits for the start of the state machine.
Wait_Sync	Waits for the PLL being is sync.
Wait_New_Outp	Wait for a new output indication.
Wait_(To_valid)	Waits for the expiration of the To_Valid timer.
Wait_To	Waits for the expiration of the To timer.
Wait_Outp	Waits for the return of the Get:output request.
Check_C-LS	Checks the controller life sign.
any_state	Any state of the above.

7.3.7.4.3.3 State machine description

This state machine drives the output of a particular isochronous submodule according to the SYNCH signal as being provided by the ISOM_SYNC state machine and according to the

T_IO_OutputValid and T_IO_Output times as being configured by the Isochronous Mode data record.

The state machine issues alarms, whenever the outputs are not being received in time (e.g. after T_IO_OutputValid) or else the outputs have not been given to the application in time (e.g. after T_IO_Output).

7.3.7.4.3.3.4 State table

Table 210 shows the states of an ISOM_OUT state machine.

Table 210 – State table ISOM_OUT

#	Current State	Event /Condition =>Action	Next State
1	Wait_Start	Start_O.ind => O_Buffer:=NIL wrong_output_upd:=0	Wait_Sync
2	Wait_Start	OTHERS => ignore	Wait_Start
3	Wait_Sync	SYNCH_O.ind => StartTimer (To_valid) StartTimer (To)	Wait_New_Outp
4	Wait_Sync	OTHERS => ignore	Wait_Sync
5	Wait_New_Outp	Local New Output_ind () /wrong_output_upd != 1 => if (wrong_output_upd>0) wrong_output_upd:= wrong_output_upd-1 // NOTE No sequence alarm is outstanding or still in trouble with the outputs	Wait_(To_valid)
6	Wait_New_Outp	Local New Output_ind () /wrong_output_upd == 1 => wrong_output_upd := 0 // NOTE output sequence trouble is over; take back diagnosis IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, NoFault)	Wait_(To_valid)
7	Wait_New_Outp	TimerExpired (To_valid) /wrong_output_upd < limit => wrong_output_upd:= wrong_output_upd+n // NOTE output sequence error; not issued since below limit	Check_C-LS
8	Wait_New_Outp	TimerExpired (To_valid) /wrong_output_upd >= limit => // NOTE Sequence Error is issued, since output sequence errors are above limit IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Check_C-LS
9	Wait_New_Outp	TimerExpired (To) /wrong_output_upd < limit => wrong_output_upd:= wrong_output_upd+n // NOTE output sequence error; not issued since below limit	Wait_Sync

#	Current State	Event /Condition =>Action	Next State
10	Wait_New_Outp	TimerExpired (To) /wrong_output_upd >= limit => // NOTE Sequence Error is issued, since output sequence errors are above limit IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Wait_Sync
11	Wait_New_Outp	OTHERS => ignore	Wait_New_Outp
12	Wait_(To_valid)	TimerExpired (To_valid) => Local Get Output_req ()	Wait_Outp
13	Wait_(To_valid)	TimerExpired (To) /wrong_output_upd < limit => wrong_output_upd:= wrong_output_upd+n // NOTE output sequence error; not issued since below limit	Wait_Sync
14	Wait_(To_valid)	TimerExpired (To) /wrong_output_upd >= limit && ALARM_PENDING == FALSE => // NOTE Sequence Error is issued, since output sequence errors are above limit IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Wait_Sync
15	Wait_(To_valid)	Local New Output_ind () /wrong_output_upd < limit => wrong_output_upd:= wrong_output_upd+n // NOTE output sequence error; not issued since below limit	Wait_(To_valid)
16	Wait_(To_valid)	Local New Output_ind () /wrong_output_upd >= limit => // NOTE Sequence Error is issued, since output sequence errors are above limit IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Wait_(To_valid)
17	Wait_(To_valid)	OTHERS => ignore	Wait_(To_valid)
18	Wait_Outp	Local Get Output_cnf () /New_Flag => O_Buffer := Subslot_Output_Data	Check_C-LS
19	Wait_Outp	Local Get Output_cnf () /!New_Flag => ignore // NOTE old value is kept	Check_C-LS
20	Wait_Outp	TimerExpired (To) /wrong_output_upd < limit => wrong_output_upd:= wrong_output_upd+n // NOTE output sequence error; not issued since below limit	Wait_Sync
21	Wait_Outp	TimerExpired (To) /wrong_output_upd >= limit => // NOTE Sequence Error is issued, since output sequence errors are above limit IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Wait_Sync
22	Wait_Outp	OTHERS => ignore	Wait_Outp

#	Current State	Event /Condition =>Action	Next State
23	Check_C-LS	/CheckC-LS () == VALID => // NOTE O_Buffer is either new or old (in case of errors)	Wait_To
24	Check_C-LS	/CheckC-LS () == INVALID => O_Buffer := substitute value IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Wait_To
25	Wait_To	TimerExpired (To) => SYNCH_OUT	Wait_Sync
26	Wait_To	Local New Output_ind () /wrong_output_upd < limit => wrong_output_upd:= wrong_output_upd+n // NOTE output sequence error; not issued since below limit	Wait_To
27	Wait_To	Local New Output_ind () /wrong_output_upd >= limit => // NOTE Sequence Error is issued, since output sequence errors are above limit IssueAlarm (IsochroneModeAlarm, OutputTimeFailure, Fault)	Wait_To
28	Wait_To	OTHERS => ignore	Wait_To
29	any_state	Stop_O.ind =>	Wait_Start

7.3.7.4.3.3.5 Functions, Macros, Timers and Variables

Table 211 contains the functions, macros, timers and variables used by ISOM_OUT state machine and their arguments and their descriptions.

Table 211 – Functions, Macros, Timers and Variables used by the ISOM_OUT

Name	Type	Function/Meaning
To	Timer	Timer to control the takeover of the data by the application.
To_valid	Timer	Timer to control the data being available from the communication.
SYNCH_OUT	Function	Pass the Output Buffer to the Output User Data
CheckC-LS	Function	Checks the controller life sign according IEC 61800-7-203:2008 6.3.12 "user data reliability". VALID and INVALID indicates the validness of the life sign according to the rules described there. If the use of the controller life sign is not intended by the submodule, VALID is returned always.
IssueAlarm	Function	The local diagnosis is updated to the current state and a change of the state is issued as alarm. The alarm confirmation is handled internally by the function. If there is a congestion of alarms, alarms may not be send if they nullify each other.
StartTimer	Function	Starts a timer
TimerExpired	Function	Indication of a timer's expiration

7.3.7.4.3.4 ISOM_IN

7.3.7.4.3.4.1 Primitive definitions

7.3.7.4.3.4.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by ISOM_IN state machine are described in the service definition and shown in Table 212.

Table 212 – Remote primitives issued or received for ISOM_IN

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.3.7.4.3.4.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by ISOM_IN state machine are described in the service definition and shown in Table 213.

Table 213 – Local primitives issued or received for ISOM_IN

Primitive	Source	Destination	Associated parameters	Functions
Local Set Input_cnf ()	FAL	ISOM_IN	see 7.3.2.5.1	Confirms Set_Input request
Local Set Input_req	ISOM_IN	FAL	see 7.3.2.5.1	Passes application input data to FAL
Start_I.ind	ISOM_SYNC	ISOM_IN	—	Start Input Data processing
Stop_I.ind	ISOM_SYNC	ISOM_IN	—	Stop Input Data processing
SYNCH_I.ind	ISOM_SYNC	ISOM_IN	Status	Trigger Signal for Input Data Processing

7.3.7.4.3.4.2 State transition diagram

Figure 39 shows the state diagram of the ISOM_IN state machine.

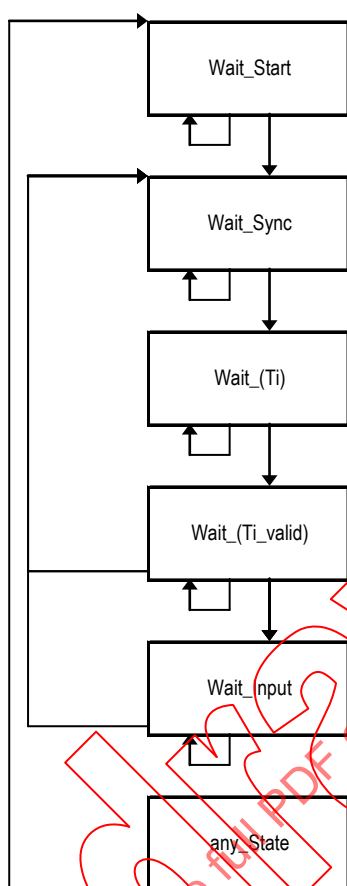


Figure 39 – State transition diagram ISOM_IN

States of the ISOM_IN state machine

Wait_Start	Waits for the start of the state machine.
Wait_Sync	Waits for the PLL being is sync.
Wait_(Ti)	Waits for the expiration of the Ti timer.
Wait_(Ti_valid)	Waits for the expiration of the Ti_valid timer.
Wait_input	Waits for the return of the Set_input request.
any_State	Any state of the above.

7.3.7.4.3.4.3 State machine description

This state machine drives the input of a particular isochronous submodule according to the SYNCH signal as being provided by the ISOM_SYNC state machine and according to the T_IO_InputValid and T_IO_Input times as being configured by the IsoM data record.

The state machine issues alarms, whenever the inputs are not being ready for transmission in time (i.e. after T_IO_InputValid).

7.3.7.4.3.4.4 State table

Table 214 shows the states of an ISOM_IN state machine.

Table 214 – State table ISOM_IN

#	Current State	Event /Condition =>Action	Next State
1	Wait_Start	Start_I.ind => SLS:=0 ALARM_PENDING := FALSE wrong_input_upd:=0	Wait_Sync
2	Wait_Start	OTHERS => ignore	Wait_Start
3	Wait_Sync	SYNCH_I.ind(Output_State) => StartTimer (Ti) StartTimer (Ti_valid) Store Output_State	Wait_(Ti)
4	Wait_Sync	OTHERS => ignore	Wait_Sync
5	Wait_(Ti)	TimerExpired (Ti) => SYNCH_IN // NOTE Fills the I_Buffer from the application	Wait_(Ti_valid)
6	Wait_(Ti)	OTHERS => ignore	Wait_(Ti)
7	Wait_(Ti_valid)	TimerExpired (Ti_valid) /CheckLS () == VALID && I_Buffer != NIL => Input_Data:= I_Buffer inc_LS(SLS) Input_Data.SLS:= SLS if (wrong_input_upd>0) wrong_input_upd= wrong_input_upd-1 if (wrong_input_upd==0) State := NoFault else State := Fault Local Set Input.req () IssueAlarm (IsochroneModeAlarm, InputTimeFailure, State)	Wait_Input
8	Wait_(Ti_valid)	TimerExpired (Ti_valid) /CheckLS () == INVALID && I_Buffer != NIL => Input_Data:= I_Buffer SLS := 0 Input_Data.SLS:= SLS if (wrong_input_upd>0) wrong_input_upd= wrong_input_upd-1 if (wrong_input_upd==0) State := NoFault else State := Fault Local Set Input.req () IssueAlarm (IsochroneModeAlarm, InputTimeFailure, State)	Wait_Input

#	Current State	Event /Condition =>Action	Next State
9	Wait_(Ti_valid)	TimerExpired (Ti_valid) /CheckLS () == NONE && I_Buffer != NIL => Input_Data:= I_Buffer if (wrong_input_upd>0) wrong_input_upd= wrong_input_upd-1 if (wrong_input_upd==0) State := NoFault else State := Fault Local Set Input.req () IssueAlarm (IsochroneModeAlarm, InputTimeFailure, State)	Wait_Input
10	Wait_(Ti_valid)	TimerExpired (Ti_valid) /I_Buffer == NIL && wrong_input_upd >= limit // NOTE Application didn't return buffer and input sequence errors above limit => IssueAlarm (IsochroneModeAlarm, InputTimeFailure, Fault)	Wait_Sync
11	Wait_(Ti_valid)	TimerExpired (Ti_valid) /I_Buffer == NIL && wrong_input_upd < limit => wrong_input_upd= wrong_input_upd+n	Wait_Sync
12	Wait_(Ti_valid)	OTHERS => ignore	Wait_(Ti_valid)
13	Wait_Input	Local Set Input.cnf () =>	Wait_Sync
14	Wait_Input	OTHERS => ignore	Wait_Input
15	any_State	Stop_I.ind =>	Wait_Start

7.3.7.4.3.4.5 Functions, Macros, Timers and Variables

Table 215 contains the functions, macros, timers and variables used by ISOM_IN state machine and their arguments and their descriptions.

Table 215 – Functions, Macros, Timers and Variables used by the ISOM_IN

Name	Type	Function/Meaning
Ti	Timer	Timer to control the takeover of the data from the application.
Ti_valid	Timer	Timer to control the data being available for the communication.
SYNCH_IN	Function	Samples and collects the Values of the Input User Data
IssueAlarm	Function	The local diagnosis is updated to the current state and a change of the state is issued as alarm. The alarm confirmation is handled internally by the function. If there is a congestion of alarms, alarms may not be send if they nullify each other.
StartTimer	Function	Starts a timer
TimerExpired	Function	Indication of a timer's expiration
SLS	Variable	Submodule Life Sign

7.3.8 Physical Device Management ASE

7.3.8.1 Overview

This ASE specifies the structure of the necessary parameters for the used protocols, e.g. IEEE 802.3, IP, UDP, RPC, or DCP, which are inherit by common ASEs. The Physical Device Management ASE manages the parameter in a common database independent of the device instances. It represents the physical device as shown in Figure A.1. The Physical Device Management ASE provides a set of services used to read, and write their values.

The attribute Station Name within the DCP group is essential for the function of an IO device and an IO controller. It shall be set by means of the DCP service DCP Set or local means. The IO device and IO controller shall store this attribute persistent. Without a Station Name (Chassis ID) the operation of an IO device is not permitted.

The attributes may be volatile or persistent. Volatile attributes are invalid after a power cycle. The values of persistent attributes remain after a power cycle.

There are two variants defined obtaining an IP address. Common to all variants is using the Station Name (Chassis ID) as the key for selecting the IP address.

The DCP protocol is mandatory and provides a service to set an IP address according to the Station Name (Chassis ID). An IO device obtains the IP address from the IO controller or IO supervisor via the DCP Set service. An IO device shall support a volatile IP address in general, and additionally a non-volatile IP address for fast start-up mode. The DCP server is only passive in this procedure.

NOTE 1 In order to support legacy implementations an additional mode with permanent IP addresses is permissible.

An IO controller obtains the IP address from the IO supervisor via the DCP Set service, local means, or DHCP. An IO controller shall support a persistent IP address. But it shall store the IP address only persistent if it has not received the IP address over DHCP.

The DHCP protocol should also be supported. It provides

- unique IP addresses for large systems and
- IP access for other applications (e.g. WEB access) to the IO device in case the IO controller is switched off.

The DHCP client requests the IP address using the selected options. The used DHCP options are provided by the DCP Set service or local means.

EXAMPLE 1 A small environment consists of one IO controller and two IO devices. Installing and configuring of a DHCP server is not appropriate in this case. Each device has got a unique station name. The IO controller identifies each IO device by means of the DCP Identify service with the filter station name and assigns its IP address with the DCP Set service.

EXAMPLE 2 A large environment consist of 27 IO controller and 64 IO devices per IO controller. The devices may be connected to different subnets. Installing and configuring of a DHCP server is appropriate in this case. Each device has got a unique station name and the use of DHCP is enabled. Each device requests its IP address from the DHCP server using the option e.g. Client Identifier = station name. The IO controller resolves the IP address of its related IO devices via DCP Identify or via DNS Get Host By Name service. The use of DCP Identify is restricted to a subnet.

The IO controller shall support DCP. Furthermore, it should support DNS and DHCP. DNS is used to provide IP addresses to:

- access IO devices behind IP routers
- use IO devices which receive their IP addresses from a DHCP server
- get an IP address for IO devices (without DHCP) if the IP address is not part of engineering information and DCP is selected to provide the IP address to the IO device

The startup procedure for an IO controller provides the following steps:

- start MAC interface
- start LLDP
- start DCP receiver, DCP sender, and DHCP client (if DHCP enabled)
- use local configured name of station or wait for it
- use local configured IP address or wait for it
- check its own IP address and station name for multiple use by means of ARP and DCP before usage (error duplicate IP address or duplicate station name)

The “normal” startup procedure for an IO device provides the following steps:

- start MAC interface
- start LLDP
- start DCP receiver and DHCP client (if DHCP enabled)
- use name of station (if persistent) or wait for it
- wait for connection establishment

NOTE 2 The IP address assignment to the IO device is typically part of the connection establishment.

The “fast” startup procedure for an IO device provides the following steps:

- start MAC interface (auto-negotiation off)
- start LLDP
- start DCP receiver and sender
- use name of station (if persistent) or wait for it
- send DCP hello if configured
- wait for connection establishment

NOTE 3 The IP address of the IO device will be typically assigned persistent; thus IP address assignment is typically not part of the connection establishment.

The IO controller provides the following steps to fast startup a related IO device:

- i) Wait for DCP hello indication, if configured
- ii) CM Connect using stored IP address

The detailed procedure for startup is provided in the protocol specification.

The Autoconfiguration Data enables a particular IO controller to utilize an IO device without an explicit configuration in an engineering system.

Furthermore, the Physical Device Management ASE defines specific object attributes and access methods to handle interface and port submodules. Besides the address resolution and neighborhood detection, these submodules contain additional attributes for global clock synchronization and RT_CLASS_3 routing information. All this attributes shall be stored non-volatile for a faster startup procedure.

7.3.8.2 Physical Device Management class specification

7.3.8.2.1 General

The Physical Device Management ASE defines one Physical Device Management object type:

7.3.8.2.2 Physical Device Management class specification

7.3.8.2.2.1 Template

A Physical Device Management object is described by the following template:

ASE:	Physical Device Management ASE
CLASS:	Physical Device Management
CLASS ID:	not used
PARENT CLASS:	IEEE 802.3, IEEE 802.1AB, IEEE 802.1D, IP suite, DNS, DHCP, SNMP, DCP, Media redundancy, PTCP
ATTRIBUTES:	
1	(m) Key Attribute: Implicit
2	(m) Attribute: List of Interfaces
2.1	(m) Attribute: Slot Number
2.2	(m) Attribute: Subslot Number
2.3	(m) Attribute: PD Interface Data Real
2.3.1	(m) Attribute: Chassis ID
2.3.2	(m) Attribute: MAC Address
2.3.3	(m) Attribute: IP Parameter Value
2.3.3.1	(m) Attribute: IP Address
2.3.3.2	(m) Attribute: Subnet Mask
2.3.3.3	(m) Attribute: Standard Gateway
2.4	(m) Attribute: PD Interface Data Adjust
2.4.1	(m) Attribute: Multiple Interface Mode
2.5	(o) Attribute: PD IR Data
2.5.1	(m) Attribute: PD IR Global Data
2.5.2	(m) Attribute: PD IR Frame Data
2.5.3	(m) Attribute: PD IR Begin End Data
2.6	(o) Attribute: PD Sync Data
2.6.1	(m) Attribute: PTCP Subdomain UUID
2.6.2	(m) Attribute: IR Data UUID
2.6.3	(m) Attribute: Reserved Interval Begin
2.6.4	(m) Attribute: Reserved Interval End
2.6.5	(m) Attribute: PLL Window
2.6.6	(m) Attribute: Sync Send Factor
2.6.7	(m) Attribute: Send Clock Factor
2.6.8	(m) Attribute: PTCP Timeout Factor
2.6.9	(m) Attribute: PTCP Takeover Timeout Factor
2.6.10	(m) Attribute: PTCP Master Startup Time
2.6.11	(m) Attribute: Sync Properties
2.6.11.1	(m) Attribute: Role
2.6.11.2	(m) Attribute: Sync Class
2.6.12	(m) Attribute: PTCP Master Priority 1
2.6.12.1	(m) Attribute: Priority
2.6.12.1	(m) Attribute: Level
2.6.13	(m) Attribute: PTCP Master Priority 2
2.6.14	(m) Attribute: PTCP Subdomain Name
2.7	(m) Attribute: PD IR Subframe Data
2.7.1	(m) Attribute: List of Subframes
2.7.1.1	(m) Attribute: Frame ID
2.7.1.2	(m) Attribute: SF IOCR Properties
2.7.1.2.1	(m) Attribute: Distributed Watchdog Factor
2.7.1.2.2	(m) Attribute: Restart Factor for Distributed Watchdog

2.7.1.2.3	(m) Attribute:	DFP Mode
2.7.1.2.4	(m) Attribute:	DFP Redundant Path Layout
2.7.1.2.5	(m) Attribute:	SFCRC16
2.7.1.2.6	(m) Attribute:	DFP Direction
2.7.1.3	(m) Attribute:	List of Subframe Data
2.7.1.2.1	(m) Attribute:	Position
2.7.1.2.2	(m) Attribute:	Data Length
2.8	(m) Attribute:	PD Interface MRP Data
2.8.1	(m) Attribute:	MRP Domain UUID
2.8.2	(m) Attribute:	MRP Role
2.8.3	(m) Attribute:	MRP Domain Name
2.8.4	(m) Attribute:	MRP Version
2.8.5	(m) Attribute:	MRP Params
2.8.6	(m) Attribute:	MRP Ring State Data
2.9	(m) Attribute:	PD Interface MRP Data Check
2.9.1	(m) Attribute:	MRP Domain UUID
2.9.2	(m) Attribute:	MRP Check
2.9.3	(m) Attribute:	Multiple Manager Check
2.9.4	(m) Attribute:	Domain UUID Check
2.10	(o) Attribute:	PD Interface FSU Data Adjust
2.10.1	(o) Attribute:	FS Hello Block
2.10.1.1	(m) Attribute:	FS Hello Mode
2.10.1.2	(m) Attribute:	FS Hello Interval
2.10.1.3	(m) Attribute:	FS Hello Retry
2.10.1.4	(m) Attribute:	FS Hello Delay
2.10.2	(o) Attribute:	Fast Startup Block
2.10.2.1	(m) Attribute:	Data
2.11	(o) Attribute:	PD NC Data Check
2.11.1	(m) Attribute:	Maintenance Demanded Drop Budget
2.11.2	(m) Attribute:	Maintenance Required Drop Budget
2.11.3	(m) Attribute:	Error Drop Budget
2.12	(m) Attribute:	List of Ports
2.12.1	(m) Attribute:	Slot Number
2.12.2	(m) Attribute:	Subslot Number
2.12.2	(m) Attribute:	Media Type
2.12.3	(m) Attribute:	Fiber Optic type
2.12.4	(m) Attribute:	Fiber Optic Cable type
2.12.5	(o) Attribute:	Fiber Optic Manufacturer Specific
2.12.5.1	(m) Attribute:	Vendor ID
2.12.5.2	(m) Attribute:	Manufacturer Specific Fiber Optic Data
2.12.6	(o) Attribute:	Power Budget Control
2.12.6.1	(m) Attribute:	Maintenance Demanded Power Budget
2.12.6.2	(m) Attribute:	Maintenance Required Power Budget
2.12.6.3	(m) Attribute:	Error Power Budget
2.12.7	(m) Attribute:	Preamble Length
2.12.8	(o) Attribute:	MRP Domain UUID
2.12.9	(m) Attribute:	Domain Boundary
2.12.9.1	(m) Attribute:	Ingress
2.12.9.2	(m) Attribute:	Egress
2.12.10	(m) Attribute:	Multicast Boundary
2.12.11	(m) Attribute:	Peer To Peer Boundary
2.12.11.1	(m) Attribute:	LLDP
2.12.11.2	(m) Attribute:	PTCP

2.12.11.3	(m) Attribute:	Time
2.12.12	(m) Attribute:	DCP Boundary
2.12.13	(o) Attribute:	PD Port Data Real
2.12.13.1	(m) Attribute:	Own Port ID
2.12.13.2	(m) Attribute:	List of Peers
2.12.13.2.1	(m) Attribute:	Peer Port ID
2.12.13.2.2	(m) Attribute:	Peer Chassis ID
2.12.13.2.3	(m) Attribute:	Line Delay
2.12.13.2.4	(m) Attribute:	Peer MAC Address
2.12.14	(m) Attribute:	MAU Type
2.12.15	(m) Attribute:	RTClass3PortStatus
2.12.16	(m) Attribute:	Link State
2.13	(o) Attribute:	PD Time Data
2.13.1	(m) Attribute:	Time Master Priority 1
2.13.2	(m) Attribute:	Time Master Priority 2
2.13.3	(m) Attribute:	Time PLL Window
3	(m) Attribute:	Write Persistence Flag

SERVICES:

1	(o) OpsService:	Read PDev Data
2	(m) OpsService:	Read PD Real Data
3	(m) OpsService:	Read PD Expected Data
4	(m) OpsService:	Read PD Interface Data Real
5	(m) OpsService:	Write PD Interface Adjust
6	(m) OpsService:	Read PD Interface Adjust
7	(o) OpsService:	Write PD IR Data
8	(o) OpsService:	Read PD IR Data
9	(o) OpsService:	Write PD Sync Data
10	(o) OpsService:	Read PD Sync Data
11	(o) OpsService:	Local Sync State Info
12	(o) OpsService:	Write PD IR Subframe Data
13	(o) OpsService:	Read PD IR Subframe Data
14	(o) OpsService:	Write PD Time Data
15	(o) OpsService:	Read PD Time Data
16	(o) OpsService:	Read PD Interface MRP Data Real
17	(o) OpsService:	Write PD Interface MRP Data Check
18	(o) OpsService:	Read PD Interface MRP Data Check
19	(o) OpsService:	Write PD Interface MRP Data Adjust
20	(o) OpsService:	Read PD Interface MRP Data Adjust
21	(o) OpsService:	Write PD Interface FSU Data Adjust
22	(o) OpsService:	Read PD Interface FSU Data Adjust
23	(o) OpsService:	Write PD NC Data Check
24	(o) OpsService:	Read PD NC Data Check
25	(m) OpsService:	Read PD Port Data Real
26	(m) OpsService:	Write PD Port Data Check
27	(m) OpsService:	Read PD Port Data Check
28	(m) OpsService:	Write PD Port Data Adjust
29	(m) OpsService:	Read PD Port Data Adjust
30	(o) OpsService:	Read PD Port FO Data Real
31	(o) OpsService:	Write PD Port FO Data Check
32	(o) OpsService:	Read PD Port FO Data Check
33	(o) OpsService:	Write PD Port FO Data Adjust
34	(o) OpsService:	Read PD Port FO Data Adjust
35	(o) OpsService:	Read PD Port MRP Data Real

- 36 (o) OpsService: Write PD Port MRP Data Adjust
- 37 (o) OpsService: Read PD Port MRP Data Adjust
- 38 (m) OpsService: Read PD Port Statistic
- 39 (o) OpsService: Write PD Combined Object Container

7.3.8.2.2.2 Attributes

Implicit

The key attribute Implicit indicates that the Physical Device Management object is implicitly addressed by the service.

List of Interfaces

This attribute list lists all interfaces related to the Physical Device Management object. It contains the following attributes:

Slot Number

This attribute contains the slot number of the module containing the interface submodule.

Attribute type: Unsigned16

Allowed Values: 0x0000 to 0x7FFF

Subslot Number

This attribute contains the subslot number of the interface.

Attribute type: Unsigned16

Allowed Values shall be used according to Table 216

Table 216 – Subslot number for interface submodules

Value (hexadecimal)	Meaning
0x8000	Interface Submodule 1
0x8100	Interface Submodule 2
0x8200	Interface Submodule 3
0x8300	Interface Submodule 4
0x8400	Interface Submodule 5
0x8500	Interface Submodule 6
0x8600	Interface Submodule 7
0x8700	Interface Submodule 8
0x8800	Interface Submodule 9
0x8900	Interface Submodule 10
0x8A00	Interface Submodule 11
0x8B00	Interface Submodule 12
0x8C00	Interface Submodule 13
0x8D00	Interface Submodule 14
0x8E00	Interface Submodule 15
0x8F00	Interface Submodule 16
0x9000	Special case system redundancy: Interface Submodule 1 on right or above delivering endnode
0x9100	Special case system redundancy: Interface Submodule 2 on right or above delivering endnode

Value (hexadecimal)	Meaning
0x9200	Special case system redundancy: Interface Submodule 3 on right or above delivering endnode
0x9300	Special case system redundancy: Interface Submodule 4 on right or above delivering endnode
0x9400	Special case system redundancy: Interface Submodule 5 on right or above delivering endnode
0x9500	Special case system redundancy: Interface Submodule 6 on right or above delivering endnode
0x9600	Special case system redundancy: Interface Submodule 7 on right or above delivering endnode
0x9700	Special case system redundancy: Interface Submodule 8 on right or above delivering endnode
0x9800	Special case system redundancy: Interface Submodule 9 on right or above delivering endnode
0x9900	Special case system redundancy: Interface Submodule 10 on right or above delivering endnode
0x9A00	Special case system redundancy: Interface Submodule 11 on right or above delivering endnode
0x9B00	Special case system redundancy: Interface Submodule 12 on right or above delivering endnode
0x9C00	Special case system redundancy: Interface Submodule 13 on right or above delivering endnode
0x9D00	Special case system redundancy: Interface Submodule 14 on right or above delivering endnode
0x9E00	Special case system redundancy: Interface Submodule 15 on right or above delivering endnode
0x9F00	Special case system redundancy: Interface Submodule 16 on right or above delivering endnode

PD Interface Data Real

This attribute contains the following attributes:

Chassis ID

This non-volatile attribute contains the Chassis ID which shall be used as Chassis ID subtype locally assigned compliant with IEEE 802.1AB. This field shall contain an VisibleString indicating the specific identifier for the particular chassis in this system. The value of the attribute is dependent on the attribute Multiple Interface Mode defined below. If Multiple Interface Mode is set to FALSE this attribute contains the NameOfStation, otherwise the content of this attribute is defined by local means.

Attribute type: VisibleString[240]

MAC Address

This attribute contains the value of the IEEE 802 MAC address of the interface.

Attribute type: OctetString[6]

IP Parameter Value

This parameter consists of the following elements:

IP Address

This attribute contains the current IP address of the interface according to IETF RFC 791.

Attribute type: Unsigned32

Subnet Mask

This attribute contains the current subnet mask according to IETF RFC 791.

Attribute type: Unsigned32

Standard Gateway

This attribute contains the current IP address of the standard gateway according to IETF RFC 791.

Attribute type: Unsigned32

PD Interface Data Adjust

This attribute contains the following attributes:

Multiple Interface Mode

This attribute controls the content of the LLDP fields ChassisID and PortID.

The value FALSE indicates that the LLDP field ChassisID contains the NameOfStation and the LLDP field PortID contains the name of the port.

The value TRUE indicates that the LLDP field ChassisID is defined by local means and the LLDP field PortID contains the name of the port + "." + the NameOfStation.

Attribute type: Boolean

Default value: FALSE

PD IR Data

This optional attribute contains the following attributes:

PD IR Global Data

This attribute contains the global IR data used by the engineering system as computation base. This attribute is inherited from parent class IEEE 802.1D.

PD IR Frame Data

This conditional list of elements shall be filled by the engineering system, if the device support IRT with IR Frame Data Elements only, as indicated by the GSD. This attribute is inherited from parent class IEEE 802.1D.

PD IR Begin End Data List

This list contains the Red Guard and the bandwidth administration information for the traffic classes RED, ORANGE and GREEN. This attribute is inherited from parent class IEEE 802.1D.

PD Sync Data

This optional attribute contains the following attributes:

PTCP Subdomain UUID

This attribute contains the unique identifier of the PTCP subdomain which is configured by the engineering system. A PTCP subdomain is a set of IO controller and/or IO devices with synchronized send clocks.

Attribute type: UUID

IR Data UUID

This attribute contains the unique identifier of the isochronous realtime project which is configured by the engineering system.

Attribute type: UUID

Reserved Interval Begin

This attribute contains the start of RT_CLASS_2 reserved bandwidth relatively to the current phase in nanoseconds.

Attribute type: Unsigned32

Allowed Values: 0 (meaning no reserved bandwidth for RT_CLASS_2 is defined)

NOTE 1 RT_CLASS_2 is usable in the context of legacy startup only. Only in that case, the value may be set to a value different from 0.

Reserved Interval End

This attribute contains the end of RT_CLASS_2 reserved bandwidth relatively to the current phase in nanoseconds.

Attribute type: Unsigned32

Allowed Values: 0 (meaning no reserved bandwidth for RT_CLASS_2 is defined)

NOTE 2 RT_CLASS_2 is usable in the context of legacy startup only. Only in that case, the value may be set to a value different from 0.

PLL Window

This attribute contains the maximum jitter of the start of the isochronous cycle with a time base of 1 ns. The value 0 indicates no PLL Window is defined. The attribute has no relevance for the Sync Master.

Attribute type: Unsigned32

Time Base: 1 ns

Sync Send Factor

This attribute contains the send interval of the sync PDUs. The Sync Send Factor shall be set in multiples of 31,25 µs.

Attribute type: Unsigned16

The values shall be in the range of 1 to 2 764 800 000.

Send Clock Factor

This attribute is defined in 7.3.9.4.2.

PTCP Timeout Factor

This attribute contains the Timeout Factor for detecting the loss of sync messages. The Time Base is the Sync Send Factor.

Allowed Values: 0x00 to 0xFF

Attribute type: Unsigned16

PTCP Takeover Timeout Factor

This attribute contains a timeout value for detection the loss of sync messages from the current sync master. The sync slave shall try to find a new sync master after the expiration of this time. The PTCP takeover timeout value shall be in the range from 32 ms to 16 352 ms for Sync ID = CLOCK and in the range from 32 ms to 2 759 400 000 ms for Sync ID = TIME.

Attribute type: Unsigned32

Time Base: Sync Send Factor * 31,25 µs

Allowed values: 1 to 511

Default value: 3

PTCP Master Startup Time

This attribute shall be the duration of the startup phase of the sync master. During this time the sync master shall check if there is a sync master with a higher clock quality in the sync domain.

Attribute type: Unsigned16

Time Base: 1 s

Allowed values: 0 to 300 s

Default value: 10

Sync Properties

This attribute contains the following attributes:

Role

This attribute contains the role of the device related to time synchronization.

Allowed values shall be according to Table 217.

Table 217 – Sync Properties Role

Meaning
Local sync
External sync: Role Sync Slave
Sync Master

Sync Class

This attribute contains the value for the Sync Class. Allowed values shall be according to Table 218.

Table 218 – Sync Class

Meaning
Test Stratum
Primary (GPS, atom clock)
Secondary (directly connected to a primary clock)
Connected to an external time signal (over boundary clocks)
Not connected to an external time signal
No clock synchronization

PTCP Master Priority 1

This attribute is used in the execution of the best master clock algorithm. Lower values take precedence.

This attribute contains the following attributes:

Priority

This attribute contains the priority.

Attribute type: Unsigned8

Allowed Values: 0x00 (if role is sync slave), 0x01-0x02 (if role is sync master)

Level

This attribute contains the level.

Attribute type: Unsigned8

Allowed Values: 0x00 (highest) - 0x07 (lowest)

PTCP Master Priority 2

This attribute is used in the execution of the best master clock algorithm. Lower values take precedence.

Attribute type: Unsigned8

Allowed Values: 0xFF

PTCP Subdomain Name

This attribute contains the name of the PTCP subdomain which is configured by the engineering system. A PTCP subdomain is a set of IO controller and/or IO devices with synchronized send clocks.

Attribute type: see Chassis ID

PD IR Subframe Data

This optional attribute lists the DFP information. It consists of the following attributes:

List of Subframes

This attribute lists all DFP subframes for which the IO device is involved. A list element includes the following attributes:

Frame ID

This parameter contains the frame ID of the related DFP IO CR.

Attribute type: Unsigned16

Allowed values: according to the Frame ID range usable for DFP as defined in Table 276

SF IOCR Properties

This attribute describes the properties for the DFP frame. It contains the following attributes:

Distributed Watchdog Factor

This attribute contains the distributed watchdog factor. For details to the distributed watchdog see Annex F.

The attribute is set to "0" only if the IO controller is able to work on subframe level.

Table 219 – Distributed Watchdog Factor

Value (hexadecimal)	Meaning	Description
0x00	Mandatory	Distributed WatchDog shall be disabled
0x01 – 0x02	Optional	The value "n" leads to an AR termination for n missed frames.
0x03 – 0x1F	Mandatory	An expiration shall be signaled to the application as a hint for a possible communication disturbance.
0x20 – 0xFF	Optional	

Time base: $\text{SendClockFactor} * \text{ReductionRatio} * 31,25 \mu\text{s}$

Attribute type: Unsigned8

Allowed values: see Table 219

Restart Factor for Distributed Watchdog

This attribute describes the restart factor for the distributed watchdog.

The attribute is set to “0” only if the IO controller is able to work on subframe level.

Table 220 – Restart Factor For Distributed Watchdog

Value (hexadecimal)	Meaning	Description
0x00	Mandatory	No restart delay necessary
0x01 – 0x09	Optional	Less than 1 s restart delay
0x0A – 0x50	Mandatory	1 s to 8 s restart delay
0x51 – 0xFF	Optional	More than 8 s restart delay

The RestartFactorForDistributedWD shall only be used if the SFIOCRProperties.DistributedWatchDogFactor is not set to zero.

NOTE 3 The RestartForDistributedWDTime is defined as minimum time having an accuracy between ‘-0’ and ‘+1 s’.

Time base: 100 ms

Attribute type: Unsigned8

Allowed values: see Table 220

DFP Mode

This attribute describes the DFP mode of the frame.

Table 221 – DFP Mode

Value (hexadecimal)	Meaning	Description
0x00	Mandatory	End node mode
0x01 – 0x3F	Mandatory	Concatenating mode Own Subframe position
0x40 – 0xFF	Optional	Concatenating mode Own Subframe position

Attribute type: Unsigned8

Allowed values: see Table 221

DFP Redundant Path Layout

This attributed describes the layout of the redundant DFP frame.

Table 222 – SFIOCRProperties.DFPRedundantPathLayout

Value (hexadecimal)	Meaning	Description
0x00	Mandatory	The Frame for the redundant path, coded by the FrameID with the least significant bit set to one, shall contain the ordering shown by SubframeData.

Value (hexadecimal)	Meaning	Description
0x01	Mandatory	The Frame for the redundant path, coded by the FrameID with the least significant bit set to one, shall contain the inverse ordering of the SubframeData.

Attribute type: Boolean

Allowed values: see Table 222

SFCRC16

This attribute indicates the validity of the subframe checksum and the subframe's cycle counter.

Table 223 – SFCRC16

Value (hexadecimal)	Description
0x00	SFCRC16 and SFCycleCounter shall be created or set to zero by the sender and not checked by the receiver.
0x01	SFCRC16 and SFCycleCounter shall be created by the sender and checked by the receiver.

Attribute type: Boolean

Allowed values: see Table 223

DFP Direction

This attribute indicates the direction of the conveyance a DFP frame.

Attribute type: Boolean

Allowed values: DFP_INBOUND, DFP_OUTBOUND

List of Subframe Data

This attribute contains the layout of the DFP frame by indicating all subframes via the following attributes:

Position

This attribute defines the position number of a particular sub frame.

Attribute type: Unsigned8

Allowed values: 0x01 to 0x7F

Data Length

This attribute defines the length of a particular sub frame.

Attribute type: Unsigned8

Allowed values: 0x01 to 0xFF

PD Interface MRP Data

This optional attribute contains the following attributes:

MRP Domain UUID

This attribute is defined in the media redundancy ASE.

MRP Role

This attribute is defined in the media redundancy ASE.

MRP Domain Name

This attribute is defined in the media redundancy ASE.

MRP Version

This attribute is defined in the media redundancy ASE.

MRP Params

This attribute is defined in the media redundancy ASE.

MRP Ring State Data

This attribute is defined in the media redundancy ASE.

PD Interface MRP Data Check

This optional attribute contains the following attributes:

MRP Domain UUID

This attribute is defined in the media redundancy ASE.

MRP Check

This parameter contains the following attributes:

Multiple Manager Check

This attribute is defined in the media redundancy ASE.

Domain UUID Check

This attribute is defined in the media redundancy ASE.

PD Interface FSU Data Adjust

This optional attribute contains the following attributes to control the interface's fast startup behavior:

FS Hello Block

This optional attribute contains the following attributes to control the DCP Hello service:

FS Hello Mode

This attribute contains the currently active model to convey the Hello service of the DCP ASE.

Attribute type: Unsigned32

Allowed values: OFF, ON_LINKUP, ON_LINKUP_AFTER_DELAY

Default value: OFF

FS Hello Interval

This attribute contains the initial interval for the second Hello service request.

Attribute type: Unsigned32

Allowed values: 30_MILLISECONDS, 50_MILLISECONDS, 100_MILLISECONDS, 300_MILLISECONDS, 500_MILLISECONDS, 1_SECOND

Default value: 30_MILLISECONDS

FS Hello Retry

This attribute contains the number of retries for the Hello request service.

Attribute type: Unsigned32

Allowed values: 1 to 15

Default value: 3

FS Hello Delay

This attribute contains the initial delay for the first Hello service request. The value is valid only if the attribute DCP Hello Mode is set to ON_LINKUP_AFTER_DELAY.

Attribute type: Unsigned32

Allowed values: NO_DELAY, 50_MILLISECONDS, 100_MILLISECONDS, 500_MILLISECONDS, 1_SECOND

Default value: NO_DELAY

Fast Startup Block

This optional attribute contains the following attributes:

Data

This attribute contains manufacturer specific fast startup data.

Attribute type: OctetString

PD NC Data Check

This optional attribute contains the following attributes:

Maintenance Demanded Drop Budget

This attribute contains the number of dropped frames that is accepted per second. If the number of dropped frames on a port per second exceeds this value, a Maintenance Demanded message is generated. The maintenance message can be deactivated.

Attribute type: Unsigned32

Allowed Values: DISABLED, Maintenance Demanded Drop Budget

Maintenance Required Drop Budget

This attribute contains the number of dropped frames that is accepted per second. If the number of dropped frames on a port per second exceeds this value, a Maintenance Required message is generated. The maintenance message can be deactivated.

Attribute type: Unsigned32

Allowed Values: DISABLED, Maintenance Required Drop Budget

Error Drop Budget

This attribute contains the number of dropped frames that is accepted per second. If the number of dropped frames on a port per second exceeds this value, a Diagnosis message is generated. The diagnosis message can be deactivated.

Attribute type: Unsigned32

Allowed Values: DISABLED, Error Drop Budget

List of Ports

This attribute list lists all ports related to the interface. It contains the following attributes:

Slot Number

This attribute contains the slot number of the module containing the port submodule.

Attribute type: Unsigned16

Allowed Values: 0x0000 to 0x7FFF

Subslot Number

This attribute contains the subslot number of the local port.

Attribute type: Unsigned16

Allowed Values shall be used according to Table 224. The value for the parameter “i” in Table 224 shall correspond with the related interface submodule. Each interface may contain up to 255 ports.

NOTE 4 The related interface module may be placed in a different slot.

Table 224 – Subslot number for port submodules

Value (hexadecimal)	Meaning
0x8i01	First Port Submodule ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)
0x8i02	Port Submodule 2 ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)
0x8i03	Port Submodule 3 ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)
...	...
0x8iFF	Port Submodule 255 ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)
0x9i01	Special case system redundancy: First Port Submodule ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)
0x9i02	Special case system redundancy: Port Submodule 2 ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)
...	...
0x9iFF	Special case system redundancy: Port Submodule 255 ($0 \leq i \leq F$; shall correspond to the subslot number of the related interface submodule)

Media Type

This attribute contains the media type of the local port.

Attribute type: Unsigned16

Allowed Values: UNKNOWN, COPPER_CABLE, FIBER_OPTIC_CABLE, RADIO_COMMUNICATION

Fiber Optic type

This attribute is present only if the port's Media Type attribute is equal to FIBER_OPTIC_CABLE. It contains the type of the fiber optic media connected to the port.

Attribute type: Unsigned32

Allowed Values shall be used according to Table 225.

Table 225 – Fiber Optic Types

Value (hexadecimal)	Meaning
0x00000000	No fiber type adjusted
0x00000001	9 µm single mode fiber
0x00000002	50 µm multi mode fiber
0x00000003	62,5 µm multi mode fiber
0x00000004	SI-POF, NA=0.5
0x00000005	SI-PCF, NA=0.36

Value (hexadecimal)	Meaning
0x00000006	LowNA-POF, NA=0.3
0x00000007	GI-POF
0x00000008	GI-PCF
0x00000009 – 0x0000007F	Reserved
0x00000080 – 0x000000FF	Vendor specific
0x00000100 – 0xFFFFFFFF	Reserved

Fiber Optic Cable type

This attribute is present only if the port's Media Type attribute is equal to FIBER_OPTIC_CABLE. It contains the type of the fiber optic cable connected to the port.

Attribute type: Unsigned32

Allowed Values shall be used according to Table 226.

Table 226 – Fiber Optic Cable Types

Value (hexadecimal)	Meaning
0x00000000	No cable specified
0x00000001	Inside/outside cable, fixed installation
0x00000002	Inside/outside cable, flexible installation
0x00000003	Outdoor cable, fixed installation
0x00000004 – 0xFFFFFFFF	Reserved

Fiber optic Manufacturer Specific

This optional attribute is present only if the port's Media Type attribute is equal to FIBER_OPTIC_CABLE and if the vendor supports vendor specific fiber optic data. This attribute contains the following attributes:

Vendor ID

This attribute shall contain the vendor ID.

Attribute type: Unsigned 16

Manufacturer Specific Fiber Optic Data

This attribute shall contain vendor specific fiber optic data

Power Budget Control

This optional attribute is present only if the port's Media Type attribute is equal to FIBER_OPTIC_CABLE and power budget control is supported by the port's hardware. This optional attribute controls the power budget. It contains the following attributes:

Maintenance Required Power Budget

This attribute contains a power budget in 0,1 dB steps that is used to generate a Maintenance Required message if the current power budget of a port is less than this value. The maintenance message can be deactivated.

Attribute type: Unsigned32

Allowed Values: DISABLED, Maintenance Required Power Budget

Maintenance Demanded Power Budget

This attribute contains a power budget in 0,1 dB steps that is used to generate a Maintenance Demanded message if the current power budget of a port is less than this value. The maintenance message can be deactivated.

Attribute type: Unsigned32

Allowed Values: DISABLED, Maintenance Demanded Power Budget

Error Power Budget

This attribute contains a power budget in 0,1 dB steps that is used to generate a Diagnosis Message if the current power budget of a port is less than this value. The diagnosis message can be deactivated.

Attribute type: Unsigned32

Allowed Values: DISABLED, Error Power Budget

Preamble Length

This attribute contains the number of preamble octets the PHY shall use before the start delimiter (SD) is send. A value of 7 preamble octets is the default (as defined for Ethernet); a value of 1 preamble octet indicates a shortening of the preamble and shall only be used, if both sending and receiving peer support preamble shortening as indicated by GSD.

Attribute type: Unsigned32

Allowed Values: 1, 7

Default Value: 7

MRP Domain UUID

This optional attribute contains the MRP Domain UUID of the port if MRP is supported and enabled. Two ports of a switch are allowed to carry this attribute at the same time. Enabling MRP on a port is done either by default (two particular ports carry the default MRP Domain UUID) or by receiving a Write MRP Port Data indication. A Read MRP Port Data addressing the MRP Domain UUID of other unconfigured MRP ports shall result in a negative response with error object non existent.

Attribute type: UUID

Domain Boundary

This attribute consists of the following attributes:

Ingress

This attribute contains the value of the ingress synchronization domain boundary. The value shall be a list of allowed values for each multicast group.

Allowed values:

BLOCK_MULTICAST_ADDRESS_GROUP_0,
PASS_MULTICAST_ADDRESS_GROUP_0,

BLOCK_MULTICAST_ADDRESS_GROUP_31,
PASS_MULTICAST_ADDRESS_GROUP_31

The multicast groups are defined for the following IEEE 802 addresses:

ADDRESS_GROUP_0: 01-0E-CF-00-04-00,
01-0E-CF-00-04-20,
01-0E-CF-00-04-40,
01-0E-CF-00-04-80

ADDRESS_GROUP_1: 01-0E-CF-00-04-01,
01-0E-CF-00-04-21,
01-0E-CF-00-04-41,
01-0E-CF-00-04-81

ADDRESS_GROUP_31: 01-0E-CF-00-04-1F,
01-0E-CF-00-04-3F,
01-0E-CF-00-04-5F,
01-0E-CF-00-04-9F

Egress

This attribute contains the value of the egress synchronization domain boundary. The allowed values shall be the same as for the Ingress attribute.

Multicast Boundary

This attribute contains the indication of the multicast boundary.

Attribute type: Unsigned32

Peer to Peer Boundary

This attribute contains the indication of peer to peer boundaries. It contains of the following attributes

LLDP

This attribute contains the indication of the LLDP boundary. If set to BLOCK, the IEEE 802.1AB shall not send LLDP frames (egress filter).

Attribute type: Unsigned8

Allowed values: PASS, BLOCK

PTCP

This attribute contains the indication of the PTCP boundary. If set to BLOCK, the Precision time control ASE shall not send PTCP_DELAY frames (egress filter).

Attribute type: Unsigned8

Allowed values: PASS, BLOCK

Time

This attribute contains the indication of the Time boundary. If set to BLOCK, the IEEE 802.1AS ASE shall not send PATH_DELAY frames (egress filter).

Attribute type: Unsigned8

Allowed values: PASS, BLOCK

DCP Boundary

This attribute contains the indication of the DCP multicast boundary.

Attribute type: Unsigned32

Allowed Values: PASS, BLOCK_HELLO, BLOCK_IDENTIFY, BLOCK_BOTH

PD Port Data Real

This attribute contains the current PD port data. It consists of the following attributes:

Own Port ID

This attribute contains the port ID of the local port. If the attribute Multiple Interface Mode is set to FALSE, the attribute Own Port ID contains the name of the port in the form "port-xyz" or "port-xyz-abcde". If the attribute Multiple Interface Mode is set to TRUE, the attribute Own Port ID contains the name of the port ("port-xyz" or "port-xyz-abcde") + "." + the NameOfStation.

Attribute type: VisibleString[240]

List of Peers

This attribute contains the actual list of peers. It consists of the following attributes:

Peer Port ID

This attribute contains the value of the peer's port ID.

Attribute type: VisibleString[240]

Peer Chassis ID

This attribute contains the value of the peer's chassis ID.

Attribute type: VisibleString[240]

Line Delay

This attribute contains the value of the line delay between the peers.

Attribute type: Unsigned32

Time Base: 1 ns

Peer MAC Address

This attribute contains the value of the peer's MAC address.

Attribute type: OctetString[6]

MAU Type

This attribute is defined in the IEEE 802.3 ASE.

RTClass3PortStatus

This attribute is defined in the IEEE 802.1AB ASE.

Link State

This attribute is defined in the IEEE 802.3 ASE.

PD Time Data

This optional attribute exists only if IEEE 802.1AS is supported. It contains the following attributes:

Time Master Priority 1

This attribute contains master priority 1 as defined within IEEE 802.1AS.

Attribute type: Unsigned8

Allowed values: 0x80 (indicating the use as master), 0xFF (indicating the use as slave)

Time Master Priority 1

This attribute contains master priority 1 as defined within IEEE 802.1AS.

Attribute type: Unsigned8

Allowed values: 0xF8

Time PLL Window

This attribute contains the maximum jitter of the master time to the local time which is accepted without setting the local time. The attribute has no relevance for the Sync Master.

Attribute type: Unsigned32

Time Base: 1 ns

Allowed values: see Table 227

Table 227 – TimePLLWindow

Value (hexadecimal)	Meaning	SyncID (Time)
0x0000	Disabled	—
0x03E8	Default	Optional (1 µs)
0x2710	Default	Optional (10 µs)
0x000186A0	Default	Mandatory (100 µs)
0x000F4240	Default	Optional (1 ms)
0x00989680	Default	Optional (10 ms)
other	Reserved	—

Write Persistence Flag

This attribute shall implement the same behavior as the Write Persistence Flag attribute of the Record Data ASE. It shall be set to TRUE forcing all Physical Device Management ASE attributes to be non-volatile at least during the parameterization phase. While operation of the AR, after Application Ready service, a Write service shall be rejected with error code State Conflict if the application has changed this attribute to FALSE after the establishment of the connection.

Attribute type: Boolean

7.3.8.3 Physical Device Management service specification

The DCP services used are described separately in the DCP specification. The DHCP, DNS, and LLDP specification do not define local service interfaces. Therefore, the implementation of the mentioned service primitives is local matter and not defined in this specification.

The following services are defined to read or write ASE attributes.

7.3.8.3.1 Read PDev Data

This confirmed service may be used to read the values of the attributes in PD IR Data, PD Sync Data, and PD IR Subframe Data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PDev Data belongs to the device; thus solely the parameter Index identifies the data record index as specified in IEC 61158-6-10.

The parameter Data has a service specific structure and meaning. Table 228 shows the specific elements of Data for this service.

Table 228 – Data elements of Read PDev Data

Parameter name	service specific elements	rsp	cnf
Data	PD IR Data	C	C(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	PD IR Global Data	M	M(=)
	IR Data UUID	M	M(=)
	Max Bridge Delay	M	M(=)
	Number of Ports	M	M(=)
	List of Port Delays	M	M(=)

Parameter name	service specific elements	rsp	cnf
	Max Port Tx Delay	M	M(=)
	Max Port Rx Delay	M	M(=)
	Max Line Rx Delay	M	M(=)
	Yellow Time	M	M(=)
	PD IR Frame Data	M	M(=)
	Frame Data Properties	M	M(=)
	List of IR Frame Data Elements	M	M(=)
	Frame Send Offset	M	M(=)
	Data Length	M	M(=)
	Reduction Ratio	M	M(=)
	Phase	M	M(=)
	Frame ID	M	M(=)
	Ethertype	M	M(=)
	Rx Port	M	M(=)
	Frame Details	M	M(=)
	Tx Port Group	M	M(=)
	Number Of Tx Port Group Array Elements	M	M(=)
	Tx Port Group Array	M	M(=)
	PD IR Begin End Data	M	M(=)
	Red Guard	M	M(=)
	Start of Red Frame ID	M	M(=)
	End of Red Frame ID	M	M(=)
	Number of Ports	M	M(=)
	List of Ports	M	M(=)
	Number of Assignments	M	M(=)
	List of Assignments	M	M(=)
	Tx Begin End Assignment	M	M(=)
	Red Orange Period Begin	M	M(=)
	Orange Period Begin	M	M(=)
	Green Period Begin	M	M(=)
	Rx Begin End Assignment	M	M(=)
	Red Orange Period Begin	M	M(=)
	Orange Period Begin	M	M(=)
	Green Period Begin	M	M(=)
	Number of Phases	M	M(=)
	List of Phases	M	M(=)
	Tx Phase Assignment	M	M(=)
	Assigned Value For Reserved Begin	M	M(=)
	Assigned Value For Orange Begin	M	M(=)
	Assigned Value For Reserved End	M	M(=)
	Rx Phase Assignment	M	M(=)
	Assigned Value For Reserved Begin	M	M(=)
	Assigned Value For Orange Begin	M	M(=)
	Assigned Value For Reserved End	M	M(=)
	PD Sync Data	C	C(=)

Parameter name	service specific elements	rsp	cnf
	PTCP Subdomain ID	M	M(=)
	Reserved Interval Begin	M	M(=)
	Reserved Interval End	M	M(=)
	PLL Window	M	M(=)
	Sync Send Factor	M	M(=)
	Send Clock Factor	M	M(=)
	PTCP Timeout Factor	M	M(=)
	PTCP Takeover Timeout Factor	M	M(=)
	PTCP Master Startup Time	M	M(=)
	Sync Properties	M	M(=)
	PTCP Master Priority 1	M	M(=)
	PTCP Master Priority 2	M	M(=)
	PTCP Subdomain Name	M	M(=)
	PD IR Subframe Data	C	C(=)
	List of Frames	M	M(=)
	Frame ID	M	M(=)
	SF IOCR Properties	M	M(=)
	Distributed Watchdog Factor	M	M(=)
	Restart Factor for Distributed Watchdog	M	M(=)
	DFP Mode	M	M(=)
	DFP Redundant Path Layout	M	M(=)
	DFP Direction	M	M(=)
	SFCRC16	M	M(=)
	List of Subframe Data	M	M(=)
	Position	M	M(=)
	Data Length	M	M(=)

Data

This parameter contains the values of the object which has been read.

PD IR Data

This conditional parameter is present only if the interface has PD IR data. It is composed of the following elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Global Data

This parameter is composed of the following list elements:

IR Data UUID

This parameter contains the value of the corresponding attribute of the ASE object.

Max Bridge Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

List of Port Delays

This parameter is composed of the following list elements:

Max Port Tx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Max Port Rx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Max Line Rx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Yellow Time

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Frame Data

This parameter is composed of the following list elements:

Frame Data Properties

This parameter contains the value of the corresponding attribute of the ASE object.

List of PD IR Frame Data Elements

This parameter is composed of the following list elements:

Frame Send Offset

This parameter contains the value of the corresponding attribute of the ASE object.

Data Length

This parameter contains the value of the corresponding attribute of the ASE object.

Reduction Ratio

This parameter contains the value of the corresponding attribute of the ASE object.

Phase

This parameter contains the value of the corresponding attribute of the ASE object.

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Ethertype

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Port

This parameter contains the value of the corresponding attribute of the ASE object.

Frame Details

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Port Group

This parameter contains the value of the corresponding attribute of the ASE object.

Number Of Tx Port Group Array Elements

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Port Group Array

This parameter contains the value of the corresponding attribute of the ASE object.

IR Begin End Data

This parameter is composed of the following elements:

Red Guard

This parameter contains the value of the corresponding attribute of the ASE object.

Start of Red Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

End of Red Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

List of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Assignments

This parameter contains the value of the corresponding attribute of the ASE object.

List of Assignments

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Begin End Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Red Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Green Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Begin End Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Red Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Green Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Phases

This parameter contains the value of the corresponding attribute of the ASE object.

List of Phases

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Phase Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Orange Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved End

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Phase Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Orange Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved End

This parameter contains the value of the corresponding attribute of the ASE object.

PD Sync Data

This conditional parameter is present only if the interface has PD sync data. It is composed of the following elements:

PTCP Subdomain ID

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval End

This parameter contains the value of the corresponding attribute of the ASE object.

PLL Window

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Send Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Send Clock Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Takeover Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Startup Time

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Properties

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Priority 1

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Priority 2

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Subdomain Name

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Subframe Data

This conditional parameter is present only if the interface has PD IR subframe data. It is composed of the following elements:

List of Frames

This parameter consists of the following elements:

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

SF IOCR Properties

This parameter consists of the following elements:

Distributed Watchdog Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Restart Factor for Distributed Watchdog

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Mode

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Redundant Path Layout

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Direction

This parameter contains the value of the corresponding attribute of the ASE object.

SFCRC16

This parameter contains the value of the corresponding attribute of the ASE object.

List of Subframe Data

This parameter consists of the following elements:

Position

This parameter contains the value of the corresponding attribute of the ASE object.

Data Length

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.2 Read PD Real Data

This confirmed service may be used to make a combined read of PD Port Data Real, PD Interface MRP Data Real, PD Port MRP Data Real, PD Port FO Data Real, PD Interface Data Real and PD Port Statistic. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Real Data belongs to the device; thus solely the parameter Index identifies the data record index as specified in IEC 61158-6-10.

The parameter Data has a service specific structure and meaning. Table 229 shows the specific elements of Data for this service.

Table 229 – Data elements of Read PD Real Data

Parameter name	service specific elements	rsp	cnf
Data	List of Submodules	M	M(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	PD Port Data Real	C	C(=)
	Own Port ID	M	M(=)
	List of Peers	C	C(=)
	Peer Port ID	M	M(=)
	Peer Chassis ID	M	M(=)
	Line Delay	M	M(=)
	Peer MAC Address	M	M(=)
	MAU type	M	M(=)
	RTClass3PortStatus	M	M(=)
	Multicast Boundary	M	M(=)
	Link State	M	M(=)
	Media Type	M	M(=)
	PD Interface MRP Data Real	C	C(=)
	MRP Domain UUID	M	M(=)
	MRP Role	M	M(=)
	MRP Domain Name	M	M(=)
	MRP Version	M	M(=)
	MRP Params	C	C(=)
	MRP Ring State Data	C	C(=)
	PD Port MRP Data Real	C	C(=)
	MRP Domain UUID	M	M(=)
	PD Port FO Data Real	C	C(=)
	Fiber Optic type	M	M(=)
	Fiber Optic Cable type	M	M(=)

Parameter name	service specific elements	rsp	cnf
	Fiber Optic Manufacturer Specific	C	C(=)
	Vendor ID	M	M(=)
	Manufacturer Specific Fiber Optic Data	M	M(=)
	Maintenance Required Power Budget	M	M(=)
	Maintenance Demanded Power Budget	M	M(=)
	Error Power Budget	M	M(=)
	PD Interface Data Real	C	C(=)
	Chassis ID	M	M(=)
	MAC Address	M	M(=)
	IP Address	M	M(=)
	PD Port Statistic	C	C(=)
	ifInOctets	M	M(=)
	ifOutOctets	M	M(=)
	ifInDiscards	M	M(=)
	ifOutDiscards	M	M(=)
	ifInErrors	M	M(=)
	ifOutErrors	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of Submodules

This parameter is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port Data Real

This conditional parameter is present only if the submodule is a port submodule. It is composed of the following elements:

Own Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

List of Peers

This conditional parameter is present only if the port has known neighbors (i.e. the port is in state LINK UP and the neighbor supports LLDP and has already delivered the LLDP information). It is composed of the following elements:

Peer Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

Peer Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

Line Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Peer MAC Address

This parameter contains the value of the corresponding attribute of the ASE object.

MAU type

This parameter contains the value of the corresponding attribute of the ASE object.

RTClass3PortStatus

This parameter contains the value of the corresponding attribute of the ASE object.

Multicast Boundary

This parameter contains the value of the corresponding attribute of the ASE object.

Link State

This parameter contains the value of the corresponding attribute of the ASE object.

Media Type

This parameter contains the value of the corresponding attribute of the ASE object.

PD Interface MRP Data Real

This conditional parameter is present only if the submodule is an interface submodule and supports MRP. It is composed of the following elements:

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Role

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Domain Name

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Version

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Params

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Ring State Data

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port MRP Data Real

This conditional parameter is present only if the submodule is a port submodule and supports MRP. It is composed of the following elements:

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port FO Data Real

This conditional parameter is present only if the submodule is a fiber optic port submodule. It is composed of the following elements:

Fiber Optic type

This parameter contains the value of the corresponding attribute of the ASE object.

Fiber Optic Cable type

This parameter contains the value of the corresponding attribute of the ASE object.

Fiber Optic Manufacturer Specific

This conditional parameter is present only if the submodule is a fiber optic port submodule. It is composed of the following elements:

Vendor ID

This parameter contains the value of the corresponding attribute of the ASE object.

Manufacturer Specific Fiber Optic Data

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Required Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

PD Interface Data Real

This conditional parameter is present only if the submodule is an interface submodule. It is composed of the following elements:

Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

MAC Address

This parameter contains the value of the corresponding attribute of the ASE object.

IP Address

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port Statistic

This conditional parameter is present only if the submodule is a port submodule. It is composed of the following elements:

ifInOctets

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifOutOctets

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifInDiscards

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifOutDiscards

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifInErrors

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifOutErrors

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

7.3.8.3.3 Read PD Expected Data

This confirmed service may be used to make a combined read of the expected data, i.e. PD Port Data Check, PD Port Data Adjust, PD Interface MRP Data Check, PD Interface MRP Data Adjust, PD Port MRP Data Adjust, PD Port FO Data Check, PD Port FO Data Adjust, PD

NC Data Check, PD Interface FSU Data Adjust and PD Interface Adjust. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Expected Data belongs to the device; thus solely the parameter Index identifies the data record index as specified in IEC 61158-6-10.

The parameter Data has a service specific structure and meaning. Table 230 shows the specific elements of Data for this service.

Table 230 – Data elements of Read PD Expected Data

Parameter name	service specific elements	rsp	cnf
Data	List of Submodules	M	M(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	PD Port Data Check	C	C(=)
	List of Peers	C	C(=)
	Peer Port ID	M	M(=)
	Peer Chassis ID	M	M(=)
	Line Delay	C	C(=)
	MAU type	C	C(=)
	Link State	C	C(=)
	Sync Difference	C	C(=)
	MAU Type Difference	C	C(=)
	PD Port Data Adjust	C	C(=)
	Adjust Domain Boundary	C	C(=)
	Adjust Multicast Boundary	C	C(=)
	Adjust MAU type	C	C(=)
	Adjust Peer to Peer Boundary	C	C(=)
	Adjust DCP Boundary	C	C(=)
	Adjust Preamble Length	C	C(=)
	PD Interface MRP Data Check	C	C(=)
	MRP Domain UUID	M	M(=)
	MRP Check	M	M(=)
	PD Interface MRP Data Adjust	C	C(=)
	MRP Domain UUID	M	M(=)
	MRP Role	M	M(=)
	MRP Domain Name	M	M(=)
	MRP Params	M	M(=)
	PD Port FO Data Check	C	C(=)
	Maintenance Required Power Budget	M	M(=)
	Maintenance Demanded Power Budget	M	M(=)
	Error Power Budget	M	M(=)
	PD Port FO Data Adjust	C	C(=)
	Adjust Fiber Optic type	M	M(=)
	Adjust Fiber Optic Cable type	M	M(=)
	PD NC Data Check	C	C(=)
	Maintenance Required Drop Budget	M	M(=)

Parameter name	service specific elements	rsp	cnf
	Maintenance Demanded Drop Budget	M	M(=)
	Error Drop Budget	M	M(=)
	PD Interface FSU Data Adjust	C	C(=)
	FS Hello Block	C	C(=)
	FS Hello Mode	M	M(=)
	FS Hello Interval	M	M(=)
	FS Hello Retry	M	M(=)
	FS Hello Delay	M	M(=)
	Fast Startup Block	C	C(=)
	Data	M	M(=)
	PD Interface Adjust	C	C(=)
	Multiple Interface Mode	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of Submodules

This parameter is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port Data Check

This conditional parameter is present only if the submodule is a port submodule and PD port data check has been written. It is composed of the following elements:

List of Peers

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Peer Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

Peer Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

Line Delay

This conditional parameter contains the value of the corresponding attribute of the ASE object.

MAU type

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Link State

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Sync Difference

This conditional parameter contains the value of the corresponding attribute of the ASE object.

MAU Type Difference

This conditional parameter contains the value of the corresponding attribute of the ASE object.

PD Port Data Adjust

This conditional parameter is present only if the submodule is a port submodule and PD port data adjust has been written. It is composed of the following elements:

Adjust Domain Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Multicast Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust MAU type

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Peer to Peer Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust DCP Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Preamble Length

This conditional parameter contains the value of the corresponding attribute of the ASE object.

PD Interface MRP Data Check

This conditional parameter is present only if the submodule is an interface submodule and PD interface MRP data check has been written. It is composed of the following elements:

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Check

This parameter contains the value of the corresponding attribute of the ASE object.

PD Interface MRP Data Adjust

This conditional parameter is present only if the submodule is an interface submodule and PD interface MRP data adjust has been written. It is composed of the following elements:

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Role

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Domain Name

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Params

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port FO Data Check

This conditional parameter is present only if the submodule is a fiber optic port submodule and PD port FO data check has been written. It is composed of the following elements:

Maintenance Required Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

PD Port FO Data Adjust

This conditional parameter is present only if the submodule is a fiber optic port submodule and PD port FO data adjust has been written. It is composed of the following elements:

Adjust Fiber Optic type

This parameter contains the value of the corresponding attribute of the ASE object.

Adjust Fiber Optic Cable type

This parameter contains the value of the corresponding attribute of the ASE object.

PD NC Data Check

This conditional parameter is present only if the submodule is an interface submodule and PD NC data check has been written. It is composed of the following elements:

Maintenance Required Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

PD Interface FSU Data Adjust

This conditional parameter is present only if the submodule is an interface submodule and PD interface FSU data adjust has been written. It is composed of the following elements:

FS Hello Block

This conditional parameter consists of the following elements:

FS Hello Mode

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Interval

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Retry

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Fast Startup Block

This conditional parameter consists of the following elements:

Data

This parameter contains the value of the corresponding attribute of the ASE object.

PD Interface Adjust

This conditional parameter is present only if the submodule is an interface submodule and PD interface adjust has been written. It is composed of the following elements:

Multiple Interface Mode

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.4 Read PD Interface Data Real

This confirmed service may be used to read the real interface data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface Data Real belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 231 shows the parameters of the service.

Table 231 – Read PD Interface Data Real

Parameter name	service specific elements	rsp	cnf
Data	Chassis ID	M	M(=)
	MAC Address	M	M(=)
	IP Parameter Value	M	M(=)
	IP Address	M	M(=)
	Subnet Mask	M	M(=)
	Standard Gateway	M	M(=)

Data

This parameter contains the values of the object which has been read.

Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

MAC Address

This parameter contains the value of the corresponding attribute of the ASE object.

IP Parameter Value

This parameter consists of the following elements:

IP Address

This parameter contains the value of the corresponding attribute of the ASE object.

Subnet Mask

This parameter contains the value of the corresponding attribute of the ASE object.

Standard Gateway

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.5 Write PD Interface Adjust

This confirmed service may be used to adjust interface data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Interface Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 232 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 232 – Data elements of Write PD Interface Adjust

Parameter name	service specific elements	req	ind
Data	Multiple Interface Mode	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Multiple Interface Mode

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.6 Read PD Interface Adjust

This confirmed service may be used to read back the adjusted interface data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 233 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to one or more port submodules. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 233 – Data elements of Read PD Interface Adjust

Parameter name	service specific elements	req	ind
Data	Multiple Interface Mode	M	M(=)

Data

This parameter contains the values of the object which has been read.

Multiple Interface Mode

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.7 Write PD IR Data

This confirmed service may be used to write the attributes of the IR Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD IR Data belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 234 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 234 – Data elements of Write PD IR Data

Parameter name	service specific elements	req	ind
Data	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	PD IR Global Data	M	M(=)
	IR Data UUID	M	M(=)
	Max Bridge Delay	M	M(=)
	Number of Ports	M	M(=)
	List of Port Delays	M	M(=)
	Max Port Tx Delay	M	M(=)
	Max Port Rx Delay	M	M(=)
	Max Line Rx Delay	M	M(=)
	Yellow Time	M	M(=)
	PD IR Frame Data	M	M(=)
	Frame Data Properties	M	M(=)
	List of PD IR Frame Data Elements	M	M(=)
	Frame Send Offset	M	M(=)
	Data Length	M	M(=)
	Reduction Ratio	M	M(=)
	Phase	M	M(=)
	Frame ID	M	M(=)
	Ethertype	M	M(=)
	Rx Port	M	M(=)
	Frame Details	M	M(=)
	Tx Port Group	M	M(=)
	Number Of Tx Port Group Array Elements	M	M(=)
	Tx Port Group Array	M	M(=)
	PD IR Begin End Data	M	M(=)
	Red Guard	M	M(=)
	Start of Red Frame ID	M	M(=)
	End of Red Frame ID	M	M(=)
	Number of Ports	M	M(=)
	List of Ports	M	M(=)
	Number of Assignments	M	M(=)
	List of Assignments	M	M(=)

Parameter name	service specific elements	req	ind
	Tx Begin End Assignment	M	M(=)
	Red Orange Period Begin	M	M(=)
	Orange Period Begin	M	M(=)
	Green Period Begin	M	M(=)
	Rx Begin End Assignment	M	M(=)
	Red Orange Period Begin	M	M(=)
	Orange Period Begin	M	M(=)
	Green Period Begin	M	M(=)
	Number of Phases	M	M(=)
	List of Phases	M	M(=)
	Tx Phase Assignment	M	M(=)
	Assigned Value For Reserved Begin	M	M(=)
	Assigned Value For Orange Begin	M	M(=)
	Assigned Value For Reserved End	M	M(=)
	Rx Phase Assignment	M	M(=)
	Assigned Value For Reserved Begin	M	M(=)
	Assigned Value For Orange Begin	M	M(=)
	Assigned Value For Reserved End	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Global Data

This parameter is composed of the following list elements:

IR Data UUID

This parameter contains the value of the corresponding attribute of the ASE object.

Max Bridge Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

List of Port Delays

This parameter is composed of the following list elements:

Max Port Tx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Max Port Rx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Max Line Rx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Yellow Time

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Frame Data

This parameter is composed of the following list elements:

Frame Data Properties

This parameter contains the value of the corresponding attribute of the ASE object.

List of PD IR Frame Data Elements

This parameter is composed of the following list elements:

Frame Send Offset

This parameter contains the value of the corresponding attribute of the ASE object.

Data Length

This parameter contains the value of the corresponding attribute of the ASE object.

Reduction Ratio

This parameter contains the value of the corresponding attribute of the ASE object.

Phase

This parameter contains the value of the corresponding attribute of the ASE object.

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Ethertype

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Port

This parameter contains the value of the corresponding attribute of the ASE object.

Frame Details

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Port Group

This parameter contains the value of the corresponding attribute of the ASE object.

Number Of Tx Port Group Array Elements

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Port Group Array

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Begin End Data

This parameter is composed of the following elements:

Red Guard

This parameter contains the value of the corresponding attribute of the ASE object.

Start of Red Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

End of Red Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

List of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Assignments

This parameter contains the value of the corresponding attribute of the ASE object.

List of Assignments

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Begin End Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Red Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Green Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Begin End Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Red Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Green Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Phases

This parameter contains the value of the corresponding attribute of the ASE object.

List of Phases

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Phase Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Orange Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved End

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Phase Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Orange Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved End

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.8 Read PD IR Data

This confirmed service may be used to read the values of the attributes in IR Data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD IR Data belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 235 shows the specific elements of Data for this service.

Table 235 – Data elements of Read PD IR Data

Parameter name	service specific elements	req	ind
Data	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	PD IR Global Data	M	M(=)
	IR Data UUID	M	M(=)
	Max Bridge Delay	M	M(=)
	Number of Ports	M	M(=)
	List of Port Delays	M	M(=)
	Max Port Tx Delay	M	M(=)
	Max Port Rx Delay	M	M(=)
	Max Line Rx Delay	M	M(=)
	Yellow Time	M	M(=)
	PD IR Frame Data	M	M(=)
	Frame Data Properties	M	M(=)
	List of PD IR Frame Data Elements	M	M(=)
	Frame Send Offset	M	M(=)
	Data Length	M	M(=)
	Reduction Ratio	M	M(=)
	Phase	M	M(=)
	Frame ID	M	M(=)
	Ethertype	M	M(=)
	Rx Port	M	M(=)
	Frame Details	M	M(=)
	Tx Port Group	M	M(=)
	Number Of Tx Port Group Array Elements	M	M(=)
	Tx Port Group Array	M	M(=)
	PD IR Begin End Data	M	M(=)
	Red Guard	M	M(=)

Parameter name	service specific elements	req	ind
	Start of Red Frame ID	M	M(=)
	End of Red Frame ID	M	M(=)
	Number of Ports	M	M(=)
	List of Ports	M	M(=)
	Number of Assignments	M	M(=)
	List of Assignments	M	M(=)
	Tx Begin End Assignment	M	M(=)
	Red Orange Period Begin	M	M(=)
	Orange Period Begin	M	M(=)
	Green Period Begin	M	M(=)
	Rx Begin End Assignment	M	M(=)
	Red Orange Period Begin	M	M(=)
	Orange Period Begin	M	M(=)
	Green Period Begin	M	M(=)
	Number of Phases	M	M(=)
	List of Phases	M	M(=)
	Tx Phase Assignment	M	M(=)
	Assigned Value For Reserved Begin	M	M(=)
	Assigned Value For Orange Begin	M	M(=)
	Assigned Value For Reserved End	M	M(=)
	Rx Phase Assignment	M	M(=)
	Assigned Value For Reserved Begin	M	M(=)
	Assigned Value For Orange Begin	M	M(=)
	Assigned Value For Reserved End	M	M(=)

Data

This parameter contains the values of the object which has been read.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Global Data

This parameter is composed of the following list elements:

IR Data UUID

This parameter contains the value of the corresponding attribute of the ASE object.

Max Bridge Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

List of Port Delays

This parameter is composed of the following list elements:

Max Port Tx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Max Port Rx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Max Line Rx Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Yellow Time

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Frame Data

This parameter is composed of the following list elements:

Frame Data Properties

This parameter contains the value of the corresponding attribute of the ASE object.

List of PD IR Frame Data Elements

This parameter is composed of the following list elements:

Frame Send Offset

This parameter contains the value of the corresponding attribute of the ASE object.

Data Length

This parameter contains the value of the corresponding attribute of the ASE object.

Reduction Ratio

This parameter contains the value of the corresponding attribute of the ASE object.

Phase

This parameter contains the value of the corresponding attribute of the ASE object.

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Ethertype

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Port

This parameter contains the value of the corresponding attribute of the ASE object.

Frame Details

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Port Group

This parameter contains the value of the corresponding attribute of the ASE object.

Number Of Tx Port Group Array Elements

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Port Group Array

This parameter contains the value of the corresponding attribute of the ASE object.

PD IR Begin End Data

This parameter is composed of the following elements:

Red Guard

This parameter contains the value of the corresponding attribute of the ASE object.

Start of Red Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

End of Red Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

List of Ports

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Assignments

This parameter contains the value of the corresponding attribute of the ASE object.

List of Assignments

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Begin End Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Red Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Green Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Begin End Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Red Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Orange Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Green Period Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Number of Phases

This parameter contains the value of the corresponding attribute of the ASE object.

List of Phases

This parameter contains the value of the corresponding attribute of the ASE object.

Tx Phase Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Orange Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved End

This parameter contains the value of the corresponding attribute of the ASE object.

Rx Phase Assignment

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Orange Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Assigned Value For Reserved End

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.9 Write PD Sync Data

This confirmed service may be used to write the attributes of the Sync Data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Sync Data belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 236 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 236 – Data elements of Write PD Sync Data

Parameter name	service specific elements	req	ind
Data	PTCP Subdomain ID	M	M(=)
	Reserved Interval Begin	M	M(=)
	Reserved Interval End	M	M(=)
	PLL Window	M	M(=)
	Sync Send Factor	M	M(=)
	Send Clock Factor	M	M(=)
	PTCP Timeout Factor	M	M(=)
	PTCP Takeover Timeout Factor	M	M(=)
	PTCP Master Startup Time	M	M(=)
	Sync Properties	M	M(=)
	PTCP Master Priority 1	M	M(=)
	PTCP Master Priority 2	M	M(=)
	PTCP Subdomain Name	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

PTCP Subdomain ID

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval End

This parameter contains the value of the corresponding attribute of the ASE object.

PLL Window

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Send Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Send Clock Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Takeover Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Startup Time

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Properties

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Priority 1

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Priority 2

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Subdomain Name

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.10 Read PD Sync Data

This confirmed service may be used to read the values of the attributes in Sync Data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Sync Data belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 237 shows the specific elements of Data for this service.

Table 237 – Data elements of Read PD Sync Data

Parameter name	service specific elements	req	ind
Data	PTCP Subdomain ID	M	M(=)
	Reserved Interval Begin	M	M(=)

Parameter name	service specific elements	req	ind
	Reserved Interval End	M	M(=)
	PLL Window	M	M(=)
	Sync Send Factor	M	M(=)
	Send Clock Factor	M	M(=)
	PTCP Timeout Factor	M	M(=)
	PTCP Takeover Timeout Factor	M	M(=)
	PTCP Master Startup Time	M	M(=)
	Sync Properties	M	M(=)
	PTCP Master Priority 1	M	M(=)
	PTCP Master Priority 2	M	M(=)
	PTCP Subdomain Name	M	M(=)

Data

This parameter contains the values of the object which has been read.

PTCP Subdomain ID

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval End

This parameter contains the value of the corresponding attribute of the ASE object.

PLL Window

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Send Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Send Clock Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Takeover Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Startup Time

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Properties

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Priority 1

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Master Priority 2

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Subdomain Name

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.11 Local Sync State Info

This local service indication shall be used to indicate any change of the synchronization state of an interface. Table 238 shows the parameters of the service.

Table 238 – Local Sync State Info

Parameter name	ind
Argument	M
Sync Data	S
Slot Number	M
Subslot Number	M
PTCP Subdomain ID	M
IR Data UUID	M
Reserved Interval Begin	M
Reserved Interval End	M
PLL Window	M
Sync Send Factor	M
Send Clock Factor	M
Sync Properties	M
Sync Frame Address	M
PTCP Timeout Factor	M
Sync Error Status	S
Slot Number	M
Subslot Number	M

Argument

The argument shall convey the service specific parameters of the service indication.

Sync Data

This selective parameter shall be present if the interface has been synchronized. It is composed of the following list elements.

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Subdomain ID

This parameter contains the value of the corresponding attribute of the ASE object.

IR Data UUID

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval Begin

This parameter contains the value of the corresponding attribute of the ASE object.

Reserved Interval End

This parameter contains the value of the corresponding attribute of the ASE object.

PLL Window

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Send Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Send Clock Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Properties

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Frame Address

This parameter contains the value of the corresponding attribute of the ASE object.

PTCP Timeout Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Sync Error Status

This selective parameter shall be present if the interface has been lost the synchronization. It is composed of the following list elements:

Slot Number

This parameter contains the value of the corresponding attribute of the ASE object.

Subslot Number

This parameter contains the value corresponding attribute of the ASE object.

7.3.8.3.12 Write PD IR Subframe Data

This confirmed service may be used to write the IR subframe data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD IR Subframe Data belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 239 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 239 – Data elements of Write PD IR Subframe Data

Parameter name	service specific elements	req	ind
Data	List of Subframes	M	M(=)
	Frame ID	M	M(=)
	SF IOCR Properties	M	M(=)
	Distributed Watchdog Factor	M	M(=)
	Restart Factor for Distributed Watchdog	M	M(=)
	DFP Mode	M	M(=)
	DFP Redundant Path Layout	M	M(=)
	SFCRC16	M	M(=)
	List of Subframe Data	M	M(=)
	Position	M	M(=)
	Length	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

List of Subframes

This parameter consists of the following elements:

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

SF IOCR Properties

This parameter consists of the following elements:

Distributed Watchdog Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Restart Factor for Distributed Watchdog

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Mode

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Redundant Path Layout

This parameter contains the value of the corresponding attribute of the ASE object.

SFCRC16

This parameter contains the value of the corresponding attribute of the ASE object.

List of Subframe Data

This parameter consists of the following elements:

Position

This parameter contains the value of the corresponding attribute of the ASE object.

Data Length

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.13 Read PD IR Subframe Data

This confirmed service may be used to read back the IR subframe data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD IR Subframe Data belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 240 shows the specific elements of Data for this service.

Table 240 – Data elements of Read PD IR Subframe Data

Parameter name	service specific elements	req	ind
Data	List of Subframes	M	M(=)
	Frame ID	M	M(=)
	SF IOCR Properties	M	M(=)
	Distributed Watchdog Factor	M	M(=)
	Restart Factor for Distributed Watchdog	M	M(=)
	DFP Mode	M	M(=)
	DFP Redundant Path Layout	M	M(=)
	SFCRC16	M	M(=)
	List of Subframe Data	M	M(=)
	Position	M	M(=)
	Length	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of Subframes

This parameter consists of the following elements:

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

SF IOCR Properties

This parameter consists of the following elements:

Distributed Watchdog Factor

This parameter contains the value of the corresponding attribute of the ASE object.

Restart Factor for Distributed Watchdog

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Mode

This parameter contains the value of the corresponding attribute of the ASE object.

DFP Redundant Path Layout

This parameter contains the value of the corresponding attribute of the ASE object.

SFCRC16

This parameter contains the value of the corresponding attribute of the ASE object.

List of Subframe Data

This parameter consists of the following elements:

Position

This parameter contains the value of the corresponding attribute of the ASE object.

Data Length

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.14 Write PD Time Data

This confirmed service may be used to adjust IEEE 802.1AS settings. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Time Data belongs to a submodule, which is addressed by

the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 241 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 241 – Data elements of Write PD Time Data

Parameter name	service specific elements	req	ind
Data	Time Master Priority 1	M	M(=)
	Time Master Priority 1	M	M(=)
	Time PLL Window	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Time Master Priority 1

This parameter contains the value for the corresponding attribute of the ASE object.

Time Master Priority 2

This parameter contains the value for the corresponding attribute of the ASE object.

Time PLL Window

This parameter contains the value for the corresponding attribute of the ASE object.

7.3.8.3.15 Read PD Time Data

This confirmed service may be used to read back the adjusted interface IEEE 802.1AS settings. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 242 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to one or more port submodules. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 242 – Data elements of Read PD Time Data

Parameter name	service specific elements	req	ind
Data	Time Master Priority 1	M	M(=)
	Time Master Priority 1	M	M(=)
	Time PLL Window	M	M(=)

Data

This parameter contains the values of the object which has been read.

Time Master Priority 1

This parameter contains the value of the corresponding attribute of the ASE object.

Time Master Priority 2

This parameter contains the value of the corresponding attribute of the ASE object.

Time PLL Window

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.16 Read PD Interface MRP Data Real

This confirmed service may be used to read back the real data attributes of an interface in the context of MRP. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface MRP Data Real belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 243 shows the specific elements of Data for this service.

Table 243 – Data elements of Read PD Interface MRP Data Real

Parameter name	service specific elements	req	ind
Data	MRP Domain UUID	M	M(=)
	MRP Role	M	M(=)
	MRP Domain Name	M	M(=)
	MRP Version	M	M(=)
	MRP Params	C	C(=)
	MRP Ring State Data	C	C(=)

Data

This parameter contains the values of the object which has been read.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Role

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Domain Name

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Version

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Params

This conditional parameter contains the value of the corresponding attribute of the ASE object.

MRP Ring State Data

This conditional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.17 Write PD Interface MRP Data Check

This confirmed service may be used to write to write check data attributes for an interface in the context of MRP. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Interface MRP Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 244 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 244 –Data elements of Write PD Interface MRP Data Check

Parameter name	service specific elements	req	ind
Data	MRP Domain UUID	M	M(=)
	MRP Check	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Check

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.18 Read PD Interface MRP Data Check

This confirmed service may be used to read check data attributes of an interface in the context of MRP. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface MRP Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 245 shows the specific elements of Data for this service.

Table 245 – Data elements of Read PD Interface MRP Data Check

Parameter name	service specific elements	req	ind
Data	MRP Domain UUID	M	M(=)
	MRP Check	M	M(=)

Data

This parameter contains the values of the object which has been read.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Check

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.19 Write PD Interface MRP Data Adjust

This confirmed service may be used to adjust data attributes of an interface in the context of MRP. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Interface MRP Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 246 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 246 – Data elements of Write PD Interface MRP Data Adjust

Parameter name	service specific elements	req	ind
Data	MRP Domain UUID	M	M(=)
	MRP Role	M	M(=)
	MRP Domain Name	M	M(=)
	MRP Params	U	U(=)

Data

This parameter contains the values of the object which may be used to write.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Role

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Domain Name

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Params

This optional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.20 Read PD Interface MRP Data Adjust

This confirmed service may be used to read back the adjusted data attributes of an interface in the context of MRP. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface MRP Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index

identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 247 shows the specific elements of Data for this service.

Table 247 – Data elements of Read PD Interface MRP Data Adjust

Parameter name	service specific elements	req	ind
Data	MRP Domain UUID	M	M(=)
	MRP Role	M	M(=)
	MRP Domain Name	M	M(=)
	MRP Params	C	C(=)

Data

This parameter contains the values of the object which has been read.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Role

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Domain Name

This parameter contains the value of the corresponding attribute of the ASE object.

MRP Params

This conditional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.21 Write PD Interface FSU Data Adjust

This confirmed service may be used to adjust the data attributes of an interface in the context of fast startup. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Interface FSU Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 248 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 248 – Data elements of Write PD Interface FSU Data Adjust

Parameter name	service specific elements	req	ind
Data	FS Hello Block	U	U(=)
	FS Hello Mode	M	M(=)
	FS Hello Interval	M	M(=)
	FS Hello Retry	M	M(=)
	FS Hello Delay	M	M(=)

Parameter name	service specific elements	req	ind
	Fast Startup Block	U	U(=)
	Data	M	M(=)

Data

This parameter contains the value of the object which may be used to write.

FS Hello Block

This parameter consists of the following elements:

FS Hello Mode

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Interval

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Retry

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Fast Startup Block

This optional parameter consists of the following elements:

Data

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.22 Read PD Interface FSU Data Adjust

This confirmed service may be used to read back the adjusted data attributes of an interface in the context of MRP. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Interface FSU Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 249 shows the specific elements of Data for this service.

Table 249 – Data elements of Read PD Interface FSU Data Adjust

Parameter name	service specific elements	rsp	cnf
Data	FS Hello Block	C	C(=)
	FS Hello Mode	M	M(=)
	FS Hello Interval	M	M(=)
	FS Hello Retry	M	M(=)
	FS Hello Delay	M	M(=)
	Fast Startup Block	C	C(=)
	Data	M	M(=)

Data

This parameter contains the values of the object which has been read.

FS Hello Block

This conditional parameter consists of the following elements:

FS Hello Mode

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Interval

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Retry

This parameter contains the value of the corresponding attribute of the ASE object.

FS Hello Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Fast Startup Block

This conditional parameter consists of the following elements:

Data

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.23 Write PD NC Data Check

This confirmed service may be used to write the check data attributes for the network components. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD NC Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning. Table 250 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed interface submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 250 – Data elements of Write PD NC Data Check

Parameter name	service specific elements	req	ind
Data	Maintenance Required Drop Budget	M	M(=)
	Maintenance Demanded Drop Budget	M	M(=)
	Error Drop Budget	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Maintenance Required Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.24 Read PD NC Data Check

This confirmed service may be used to read back the check attributes for the network component data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD NC Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 251 shows the specific elements of Data for this service.

Table 251 – Data elements of Read PD NC Data Check

Parameter name	service specific elements	req	ind
Data	Maintenance Required Drop Budget	M	M(=)
	Maintenance Demanded Drop Budget	M	M(=)
	Error Drop Budget	M	M(=)

Data

This parameter contains the values of the object which has been read.

Maintenance Required Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Drop Budget

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.25 Read PD Port Data Real

This confirmed service may be used to read real port data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port Data Real belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 252 shows the specific elements of Data for this service.

Table 252 – Data elements of Read PD Port Data Real

Parameter name	service specific elements	rsp	cnf
Data	Own Port ID	M	M(=)
	List of Peers	C	C(=)
	Peer Port ID	M	M(=)
	Peer Chassis ID	M	M(=)
	Line Delay	M	M(=)

Parameter name	service specific elements	rsp	cnf
	Peer MAC Address	M	M(=)
	MAU type	M	M(=)
	RTClass3PortStatus	M	M(=)
	Multicast Boundary	M	M(=)
	Link State	M	M(=)
	Media Type	M	M(=)

Data

This parameter contains the values of the object which has been read.

Own Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

List of Peers

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Peer Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

Peer Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

Line Delay

This parameter contains the value of the corresponding attribute of the ASE object.

Peer MAC Address

This parameter contains the value of the corresponding attribute of the ASE object.

MAU type

This parameter contains the value of the corresponding attribute of the ASE object.

RTClass3PortStatus

This parameter contains the value of the corresponding attribute of the ASE object.

Multicast Boundary

This parameter contains the value of the corresponding attribute of the ASE object.

Link State

This parameter contains the value of the corresponding attribute of the ASE object.

Media Type

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.26 Write PD Port Data Check

This confirmed service may be used to enable port data to be checked. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Port Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 253 shows the specific elements of Data for this service.

Table 253 – Data elements of Write PD Port Data Check

Parameter name	service specific elements	req	ind
Data	List of Peers	U	U(=)
	Peer Port ID	M	M(=)
	Peer Chassis ID	M	M(=)
	Line Delay	U	U(=)
	MAU type	U	U(=)
	Link State	U	U(=)
	Sync Difference	U	U(=)
	MAU Type Difference	U	U(=)

Data

This parameter contains the values of the object which may be used to write.

List of Peers

This optional parameter contains the value of the corresponding attribute of the ASE object.

Peer Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

Peer Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

Line Delay

This optional parameter contains the value of the corresponding attribute of the ASE object.

MAU type

This optional parameter contains the value of the corresponding attribute of the ASE object.

Link State

This optional parameter contains the value of the corresponding attribute of the ASE object.

Sync Difference

This optional parameter contains the value of the corresponding attribute of the ASE object.

MAU Type Difference

This optional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.27 Read PD Port Data Check

This confirmed service may be used to read the check data of the port. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 254 shows the specific elements of Data for this service.

Table 254 – Data elements of Read PD Port Data Check

Parameter name	service specific elements	rsp	cnf
Data	List of Peers	C	C(=)
	Peer Port ID	M	M(=)
	Peer Chassis ID	M	M(=)
	Line Delay	C	C(=)
	MAU type	C	C(=)
	Link State	C	C(=)
	Sync Difference	C	C(=)
	MAU Type Difference	C	C(=)

Data

This parameter contains the values of the object which has been read.

List of Peers

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Peer Port ID

This parameter contains the value of the corresponding attribute of the ASE object.

Peer Chassis ID

This parameter contains the value of the corresponding attribute of the ASE object.

Line Delay

This conditional parameter contains the value of the corresponding attribute of the ASE object.

MAU type

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Link State

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Sync Difference

This conditional parameter contains the value of the corresponding attribute of the ASE object.

MAU Type Difference

This conditional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.28 Write PD Port Data Adjust

This confirmed service may be used to adjust port data. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Port Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 255 shows the specific elements of Data for this service.

Table 255 – Data elements of Write PD Port Data Adjust

Parameter name	service specific elements	req	ind
Data	Adjust Domain Boundary	U	U(=)
	Adjust Multicast Boundary	U	U(=)
	Adjust MAU type	U	U(=)
	Adjust Peer to Peer Boundary	U	U(=)
	Adjust DCP Boundary	U	U(=)
	Adjust Preamble Length	U	U(=)

Data

This parameter contains the values of the object which may be used to write.

Adjust Domain Boundary

This optional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Multicast Boundary

This optional parameter contains the value of the corresponding attribute of the ASE object.

Adjust MAU type

This optional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Peer to Peer Boundary

This optional parameter contains the value of the corresponding attribute of the ASE object.

Adjust DCP Boundary

This optional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Preamble Length

This optional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.29 Read PD Port Data Adjust

This confirmed service may be used to read adjusted port data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 256 shows the specific elements of Data for this service.

Table 256 – Data elements of Read PD Port Data Adjust

Parameter name	service specific elements	rsp	cnf
Data	Adjust Domain Boundary	C	C(=)
	Adjust Multicast Boundary	C	C(=)
	Adjust MAU type	C	C(=)

Parameter name	service specific elements	rsp	cnf
	Adjust Peer to Peer Boundary	C	C(=)
	Adjust DCP Boundary	C	C(=)
	Adjust Preamble Length	C	C(=)

Data

This parameter contains the values of the object which has been read.

Adjust Domain Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Multicast Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust MAU type

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Peer to Peer Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust DCP Boundary

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Adjust Preamble Length

This conditional parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.30 Read PD Port FO Data Real

This confirmed service may be used to read real fiber optic data. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1 4.1.1). PD Port Data Real belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 257 shows the specific elements of Data for this service.

Table 257 – Data elements of Read Port FO Data Real

Parameter name	service specific elements	rsp	cnf
Data	Fiber Optic type	M	M(=)
	Fiber Optic Cable type	M	M(=)
	Fiber Optic Manufacturer Specific	C	C(=)
	Vendor ID	M	M(=)
	Manufacturer Specific Fiber Optic Data	M	M(=)
	Maintenance Required Power Budget	M	M(=)
	Maintenance Demanded Power Budget	M	M(=)

Parameter name	service specific elements	rsp	cnf
	Error Power Budget	M	M(=)

Data

This parameter contains the values of the object which has been read.

Fiber Optic type

This parameter contains the value of the corresponding attribute of the ASE object.

Fiber Optic Cable type

This parameter contains the value of the corresponding attribute of the ASE object.

Fiber Optic Manufacturer Specific

This conditional parameter contains the value of the corresponding attribute of the ASE object.

Vendor ID

This parameter contains the value of the corresponding attribute of the ASE object.

Manufacturer Specific Fiber Optic Data

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Required Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.31 Write PD Port FO Data Check

This confirmed service may be used to write check data attributes for a fiber optic port. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Port FO Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 258 shows the specific elements of Data for this service.

Table 258 – Data elements of Write PD Port FO Data Check

Parameter name	service specific elements	req	ind
Data	Maintenance Required Power Budget	M	M(=)
	Maintenance Demanded Power Budget	M	M(=)
	Error Power Budget	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Maintenance Required Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.32 Read PD Port FO Data Check

This confirmed service may be used to read the check data attributes for a fiber optic port. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port FO Data Check belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 259 shows the specific elements of Data for this service.

Table 259 – Data elements of Read PD Port FO Data Check

Parameter name	service specific elements	rsp	cnf
Data	Maintenance Required Power Budget	M	M(=)
	Maintenance Demanded Power Budget	M	M(=)
	Error Power Budget	M	M(=)

Data

This parameter contains the values of the object which has been read.

Maintenance Required Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Maintenance Demanded Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

Error Power Budget

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.33 Write PD Port FO Data Adjust

This confirmed service may be used to adjust data attributes for a fiber optic port. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Port FO Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 260 shows the specific elements of Data for this service.

Table 260 – Data elements of Write PD Port FO Data Adjust

Parameter name	service specific elements	req	ind
Data	Adjust Fiber Optic type	M	M(=)
	Adjust Fiber Optic Cable type	M	M(=)

Data

This parameter contains the values of the object which may be used to write.

Adjust Fiber Optic type

This parameter contains the value of the corresponding attribute of the ASE object.

Adjust Fiber Optic Cable type

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.34 Read PD Port FO Data Adjust

This confirmed service may be used to read the adjusted data attributes for a fiber optic port. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port FO Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 261 shows the specific elements of Data for this service.

Table 261 – Data elements of Read PD Port FO Data Adjust

Parameter name	service specific elements	rsp	cnf
Data	Adjust Fiber Optic type	M	M(=)
	Adjust Fiber Optic Cable type	M	M(=)

Data

This parameter contains the values of the object which has been read.

Adjust Fiber Optic type

This parameter contains the value of the corresponding attribute of the ASE object.

Adjust Fiber Optic Cable type

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.35 Read PD Port MRP Data Real

This confirmed service may be used to read the data attributes of a port in the context of MRP. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port MRP Data Real belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 262 shows the specific elements of Data for this service.

Table 262 – Data elements of Read PD Port MRP Data Real

Parameter name	service specific elements	rsp	cnf
Data	MRP Domain UUID	M	M(=)

Data

This parameter contains the value of the object which has been read.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.36 Write PD Port MRP Data Adjust

This confirmed service may be used to write to adjust data attributes of a port in the context of MRP. This service shall be used in conjunction with the IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the write service (see 7.3.1.4.2.1). PD Port MRP Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 263 shows the specific elements of Data for this service.

The service user shall only use this service if the used AR is connected to the addressed port submodule. Otherwise the service provider shall issue a negative response to the service and all data shall be ignored.

Table 263 – Data elements of Write PD Port MRP Data Adjust

Parameter name	service specific elements	req	ind
Data	MRP Domain UUID	M	M(=)

Data

This parameter contains the value of the object which may be used to write.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.37 Read PD Port MRP Data Adjust

This confirmed service may be used to read back adjusted data attributes of a port in the context of MRP. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port MRP Data Adjust belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 264 shows the specific elements of Data for this service.

Table 264 – Data elements of Read PD Port MRP Data Adjust

Parameter name	service specific elements	rsp	cnf
Data	MRP Domain UUID	M	M(=)

Data

This parameter contains the values of the object which has been read.

MRP Domain UUID

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.8.3.38 Read PD Port Statistic

This confirmed service may be used to read the port statistic counters. This service shall be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). PD Port Statistic belongs to a submodule, which is addressed by the default API (zero), its slot and subslot number. The parameter Index identifies the data record index as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface or a port an error is returned.

The parameter Data has a service specific structure and meaning. Table 265 shows the specific elements of Data for this service.

Table 265 – Data elements of Read PD Port Statistic

Parameter name	service specific elements	rsp	cnf
Data	ifInOctets	M	M(=)
	ifOutOctets	M	M(=)
	ifInDiscards	M	M(=)
	ifOutDiscards	M	M(=)
	ifInErrors	M	M(=)
	ifOutErrors	M	M(=)

Data

This parameter contains the values of the object which has been read.

ifInOctets

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifOutOctets

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifInDiscards

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifOutDiscards

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifInErrors

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

ifOutErrors

This parameter contains the value of the corresponding attribute of the SNMP ASE object.

7.3.8.3.39 Write PD Combined Object Container

This confirmed service may be used to adjust the physical device settings with a combined object container. This service shall be used in conjunction with the IO AR or Supervisor AR.

For the physical device the combined object consists of

- An interface submodule, acting as the combined object container's access point
- Port submodules related to this interface submodule

The parameter structure and the meaning of the parameters is the same as described in the write combined object service (see 7.3.1.4.2.2). PD Combined Object Container belongs to all submodules related to a particular interface submodule (i.e. the interface submodule and all ports related to it), which is addressed by the default API (zero), the slot and subslot number of the interface submodule. The parameter Index identifies the data record index for the combined object container as specified in IEC 61158-6-10. If the addressed submodule does not represent an interface an error is returned.

The parameter Data has a service specific structure and meaning according to 7.3.1.4.2.2. Table 113 shows the specific elements of Data for this service.

The PD Combined Object Container acts quasi as a script issuing a bunch of write services. Each container element represents a particular write service, where the container element attributes API, Slot Number and Subslot Number address the submodule being target of the container element (i.e. interface submodule or some port submodule), index addresses the service to be applied and Data holds the data to be written.

The following services are supported to be integrated into the PD Combined Object Container. To ease the description, only a reference to the service description is given.

A particular container element addressing the interface submodule may contain one of the following services:

- Write PD Interface Adjust (see 7.3.8.3.5)
- Write PD IR Data (see 7.3.8.3.7)
- Write PD Sync Data (see 7.3.8.3.9)
- Write PD IR Subframe Data (see 7.3.8.3.12)
- Write PD Time Data (see 7.3.8.3.14)
- Write PD Interface MRP Data Check (see 7.3.8.3.17)
- Write PD Interface MRP Data Adjust (see 7.3.8.3.19)
- Write PD Interface FSU Data Adjust (see 7.3.8.3.21)
- Write PD NC Data Check (see 7.3.8.3.23)

A particular container element addressing a port submodule may contain one of the following services:

- Write PD Port Data Check (see 7.3.8.3.26)
- Write PD Port Data Adjust (see 7.3.8.3.28)
- Write PD Port FO Data Check (see 7.3.8.3.31)
- Write PD Port FO Data Adjust (see 7.3.8.3.33)
- Write PD Port MRP Data Adjust (see 7.3.8.3.36)

7.3.9 AR ASE

7.3.9.1 Overview

7.3.9.1.1 General

This ASE specifies the structure of the ARs including the necessary DL parameters for the specific CRs. The AR ASE in each device manages an end point of the AR and all related CRs, and maintains their local context. The AR endpoint is related to a dedicated list of application processes, which is addressed during the connection establishment phase, and all further services are within this context.

Furthermore, the application layer defines an implicit AR which does not have a special AR ASE entry. The implicit AR is used for all remote Context Management services to establish and release dedicated IO or Supervisor ARs and for read access to other APOs without establishing a dedicated AR. Write access is not permitted over the implicit AR. This AR is locally addressed by an implicit AREP. The implicit AR requires addressing of the application process via API read service primitive parameter.

An AR is characterized by all of the following basic roles of the AREP used:

AR cardinality

one-to-one

- In a one-to-one AR an AP communicates with exactly one remote AP.

one-to-many

- In a one-to-many AR the AP of an IO device communicates with the APs of a group of IO controller and IO devices. Exactly one AP is addressed per device.

Device type relations

IO communication

- between IO controller and IO devices
- between IO controller and other IO controller
- between IO devices and IO devices (at least one IO Controller shall be involved to configure and monitor the relationship between IO devices)

Supervisor communication

- between IO supervisor and IO devices

Implicit communication

- between IO controller and IO devices
- between IO supervisor and IO devices

Each IO device shall provide one implicit AR that allows processing at least one outstanding service all the time. It may be able to process more services in parallel according to local resources.

Each IO device shall provide resources for at least one explicit AR. An IO device may be able to provide more ARs in parallel according to local resources.

7.3.9.1.2 Communication relationship

7.3.9.1.2.1 General

An AR is mapped to one or more communication relationships (CRs). A CR carries out the exchange of information based on a set of rules.

7.3.9.1.2.2 Communication relationship endpoint

CRs are defined as a set of co-operating Communication Relationship End Points (CREPs). The local context of a CR endpoint is used to control the conveyance of APDUs on the CR. Figure 40 illustrates the Assignment of Communication Relationship to Application Relationship.

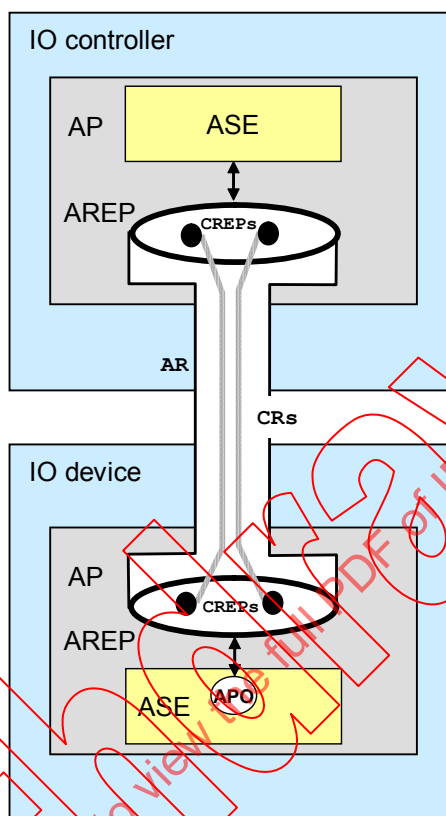


Figure 40 – Assignment of communication relationship to application relationship

7.3.9.1.2.3 Buffer and queues

CREPs may be modeled as a queue or as a buffer. Application data transferred over a queued CREP are delivered in the order received for conveyance. The transfer of application data over a buffered CREP is different. In this case, application data is placed in an application layer buffer for transfer.

When the CREP receives another conveyance request, it replaces the previous contents of the buffer whether or not transmitted. Once application data are written into a buffer for transfer, it is preserved in the buffer until the next application data to be transmitted replaces it. While in the buffer, application data may be read more than once without deleting it from the buffer or changing its contents.

At the receiving end, the operation is similar. The receiving application layer places the received application data into a buffer for access by the different ASEs. When subsequent application data are received, it overwrites the previous application data in the buffer whether or not it was read by the corresponding ASE. Reading the application data from the buffer is not destructive it does not destroy or change the contents of the buffer, allowing the contents to be read from the buffer one or more times.

7.3.9.1.2.4 Communication relationship endpoint roles

A communication relationship is characterized by all of the following basic roles of the CREP used:

CR cardinality

one-to-one:

In a one-to-one communication relationship a CREP communicates with exactly one remote CREP.

one-to-many:

In a one-to-many relationship one CREP communicates simultaneously with the CREPs of a group of devices. A one-to-many communication relationship does not permit confirmation of the execution of a request.

Connection mode

connection-oriented

In a connection-oriented application relationship a logical connection is established between application processes. Three different phases are identified in connection-oriented ARs:

- connection establishment phase
- data transfer phase and
- connection release phase

Exchanging data on a connection shall only be possible after a successful connection establishment. In the connection establishment phase a request to establish the connection is announced at the remote application process (e.g. Connect service). The request to establish the connection-oriented AR contains the information about the context to be used in the data transfer phase. The remote application informs the requesting application process about the acceptance of the context. If the context is accepted both application processes are in the data transfer phase and are able to communicate with each other according to the agreed context. Connection-oriented AR can be monitored. A connection-oriented AR is dissolved by its release (e.g. Release service).

connectionless

Connectionless application relationships have neither a connection establishment nor a connection release. They are always in the data transfer phase. It is not possible to monitor the connectionless AR.

Transmission types

cyclic

The transmission type "cyclic" means that after an initial request by the application process the application relationship is established and afterwards the data transfer is executed periodically.

acyclic

The transmission type "acyclic" means that a single transfer is executed after each request by the application process.

both

The transmission type "both" means that a cyclic as well as acyclic transfer is supported.

Transmission characteristics

Buffer to Buffer

The transmission is characterized by a buffer to buffer transfer.

Queue to Queue

The transmission is characterized by a queue to queue transfer.

Buffer to Queue

The transmission is characterized by a buffer to queue transfer.

Queue to Buffer

The transmission is characterized by a queue to buffer transfer.

Service types**Confirmed**

Only confirmed services are allowed.

Unconfirmed

Only unconfirmed services are allowed.

Transmission directions:**Bi-directional**

APDUs are transmitted in both directions.

Unidirectional

APDUs are transmitted only in one direction.

Service relations**Client-Server communication**

The services are used according to the Client/Server model.

Producer-Consumer communication

The services are used according to the Producer/Consumer model.

7.3.9.1.2.5 Overview of communication relationships

The application layer uses the following types of communication relationships:

BBUU (Buffer Buffer Unconfirmed Unidirectional)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connection-oriented
- cyclic
- buffer to buffer
- unconfirmed
- unidirectional
- Producer/Consumer

BBUB (Buffer Buffer Unconfirmed Bidirectional)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connection-oriented
- cyclic
- buffer to buffer
- unconfirmed
- bi-directional
- Client/Server

BBUU-OM (Buffer Buffer Unconfirmed Unidirectional - One to Many)

This type of communication relationship is characterized by the following role attributes:

- one to many
- connection-oriented
- cyclic
- buffer to buffer

- unconfirmed
- unidirectional
- Producer/Consumer

QQCB-CO (Queue Queue Confirmed Bi-directional - Connection-Oriented)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connection-oriented
- acyclic
- queue to queue
- confirmed
- bi-directional
- Client/Server

QQCB-CL (Queue Queue Confirmed Bi-directional - Connectionless)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connectionless
- acyclic
- queue to queue
- confirmed
- bi-directional
- Client/Server

7.3.9.2 Application relationships

7.3.9.2.1 Implicit application relationship

The Implicit AR reflects the application relationship (see Figure 41) between the application processes of one IO controller and one IO device, and one IO supervisor and one IO device. Furthermore, the Implicit AR provides the application relationship for the following purposes:

- acyclic data transfer for general purpose data access (read) e.g. identification, parameterization, and diagnosis

NOTE Write access is only allowed via explicit established IO or Supervisor ARs.

The Implicit AR contains the following CR type:

- QQCB-CL to access general purpose data (read), diagnosis and identification information

Each IO device shall provide one implicit AR that allows processing at least one outstanding service all the time. It may be able to process more services in parallel according to local resources.

The implicit AR contains one implicit CR. The endpoints AREP and CREP are unique and implicit addressed by local means. The AR UUID of the implicit AR shall be zero.

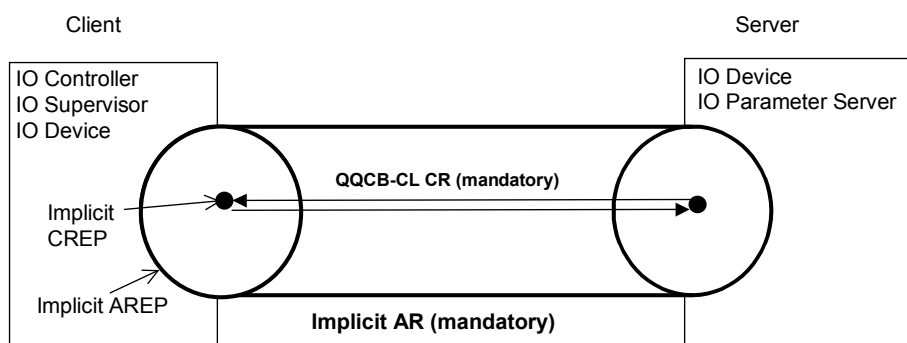


Figure 41 – Implicit application relationship

7.3.9.2.2 IO application relationship

The IO AR reflects the application relationship (see Figure 42) between the application processes of one IO controller and one AP of an IO device or more IO devices in case input data are published. Furthermore, the IO AR model supports redundancy and reconfiguration during operation by means of using primary and backup IO ARs. The IO AR provides the application relationship for the following purposes:

- cyclic exchange of output data, optional as multicast
- cyclic exchange of input data, optional as multicast
- cyclic exchange of provider and consumer status information
- acyclic data transfer for general purpose data access e.g. identification, parameterization, and diagnosis
- report and acknowledgment of alarms

The IO AR contains the following CR types:

- at least one BBUU (mandatory) to transmit input data with its provider status from the IO device (provider) to the IO controller (consumer) and/or to transmit consumer status information (IOCS) for output data from the IO device to the IO controller or alternatively one BBUU-OM (optional) to transmit input data with its provider status from the IO device (publisher) to several IO controller (consumer) and several IO devices (consumer)
- at least one BBUU (mandatory) to transmit output data with its provider status from the IO controller (provider) to the IO device (consumer) and/or to transmit consumer status information (IOCS) for input data from the IO controller to the IO device or alternatively one BBUU-OM (optional) to transmit output data with its provider status from the IO controller (publisher) to several IO devices (consumer) and to several IO controller (consumer)
- one QQCB (mandatory) to transmit alarms from an IO device as source to an IO controller as destination and acknowledge in the opposite direction. It can further be used to transmit alarms from an IO controller as source to an IO device as destination and acknowledge in the opposite direction
- one QQCB-CO (mandatory) to access general purpose data (read or write), diagnosis and identification information

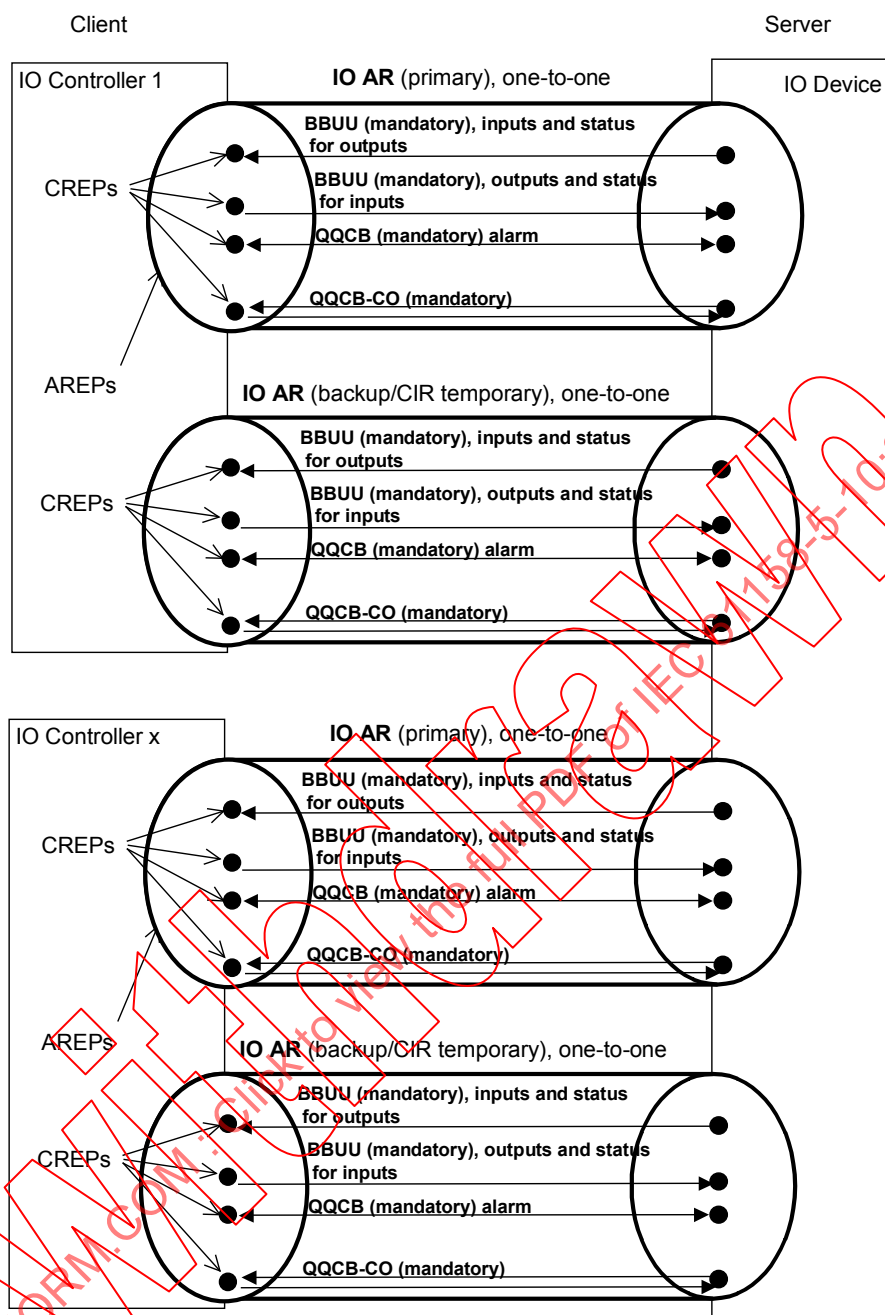


Figure 42 – Example IO application relationship (one-to-one)

- Figure 43 illustrates the IO AR with one M CR for one to many uses.

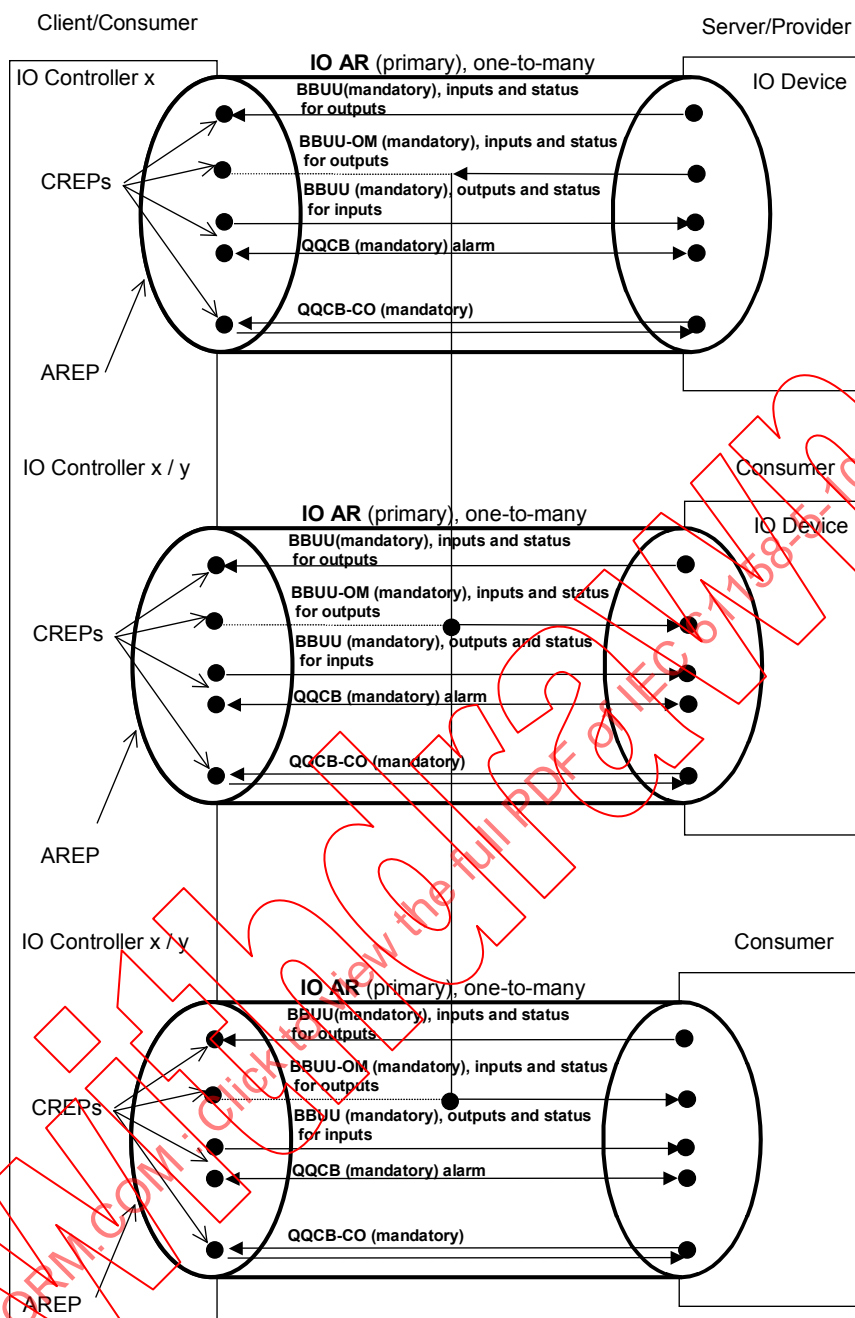


Figure 43 – Example IO application relationship one-to-many

The IO AR has two facets in relation to the startup behavior:

- normal mode
- fast startup mode, to shorten the time for connection establishment

The IO device shall support the advanced startup mode and may support the legacy startup mode.

7.3.9.2.3 Supervisor application relationship

The Supervisor AR has two facets:

- force outputs as specialized IO AR

- device access for device parameterization

In the first case, the Supervisor AR is a specialized IO AR where the attribute AR type is set to IOS AR. This kind of AR allows to takeover a submodule if allowed. An IO AR may prevent this supervisor takeover with the attribute Takeover Allowed is set to NOT_ALLOWED.

For device access, the Supervisor AR shall only use the QQCB-CO to access general purpose data (read or write) without restrictions accessing slots/subslots of the application.

7.3.9.3 ARL class specification

7.3.9.3.1 Template

The ARL object is described by the following template:

ASE: AR ASE

CLASS: ARL

CLASS ID: not used

PARENT CLASS: TOP

ATTRIBUTES:

1	(m)	Key Attribute: Implicit
2	(m)	Attribute: List of ARs
2.1	(m)	Attribute: AREP
2.2	(m)	Attribute: AR type
2.3	(m)	Attribute: AR UUID
2.4	(m)	Attribute: Session Key
2.5	(m)	Attribute: CM Initiator MAC Add
2.6	(m)	Attribute: CM Responder MAC Add
2.7	(m)	Attribute: CM Initiator Object UUID
2.8	(m)	Attribute: AR Properties
2.8.1	(m)	Attribute: State
2.8.1	(m)	Attribute: Supervisor Takeover Allowed
2.8.2	(m)	Attribute: Parameterization Server
2.8.3	(m)	Attribute: Device Access
2.8.4	(m)	Attribute: Companion AR
2.8.5	(m)	Attribute: Acknowledge Companion AR
2.8.6	(m)	Attribute: Startup Mode
2.8.7	(o)	Attribute: Pull Module Alarm Allowed
2.9	(m)	Attribute: CM Initiator Activity Timeout Factor
2.10	(m)	Attribute: Remote Application Ready Timeout
2.11	(m)	Attribute: Initiator UDP RT Port
2.12	(m)	Attribute: Responder UDP RT Port
2.13	(m)	Attribute: CM Initiator Station Name
2.14	(m)	Attribute: List of CRs
2.14.1	(m)	Attribute: CREP
2.15	(o)	Attribute: Parameter Server Block
2.15.1	(m)	Attribute: Parameter Server Object UUID
2.15.2	(m)	Attribute: Parameter Server Properties
2.15.3	(m)	Attribute: CM Initiator Activity Timeout Factor
2.15.4	(m)	Attribute: Parameter Server Station Name
2.16	(m)	Attribute: AR RPC Server Block
2.16.1	(m)	Attribute: Initiator RPC Server Port
2.16.2	(m)	Attribute: Responder RPC Server Port
2.17	(o)	Attribute: IR Info Block
2.17.1	(m)	Attribute: IR Data UUID

2.17.2	(m) Attribute:	Number of IO CRs
2.17.3	(o) Attribute:	List of DFP IO CRs
2.17.3.1	(m) Attribute:	IO CR Reference
2.17.3.2	(m) Attribute:	Sub Frame Offset
2.17.3.3	(m) Attribute:	Sub Frame Data
2.17.3.3.1	(m) Attribute:	Sub Frame Position
2.17.3.3.2	(m) Attribute:	Sub Frame Data Length
2.18	(o) Attribute:	SR Info Block
2.18.1	(m) Attribute:	Redundancy Data Hold Factor
2.18.2	(m) Attribute:	SR Properties
2.18.2.1	(m) Attribute:	Input Valid on Backup AR
2.18.2.2	(m) Attribute:	Activate Redundancy Alarm
2.19	(o) Attribute:	Redundancy Info
2.19.1	(m) Attribute:	End Point 1
2.19.2	(m) Attribute:	End Point 2
2.20	(o) Attribute:	List of AR Vendor Blocks
2.20.1	(m) Attribute:	AP Structure Identifier
2.20.2	(m) Attribute:	API
2.20.3	(m) Attribute:	Data Request
2.20.4	(m) Attribute:	Data Response
2.21	(o) Attribute:	Fast Startup Mode
2.22	(m) Attribute:	AR State

SERVICES:

1	(m) OpsService:	Read AR Data for one API
2	(m) OpsService:	Read AR Data for one device
3	(m) OpsService:	Local Set AR State
4	(m) OpsService:	Local AR In Data

7.3.9.3.2 Attributes**Implicit**

The key attribute Implicit indicates that the ARL object is implicitly addressed by the service.

List of ARs

This attribute list contains the following attributes:

AREP

This attribute is the local identifier for the AR entry.

AR type

This attribute defines the type of the AR entry.

If the device supports dynamic reconfiguration during operation or system redundant operation than it shall accept the AR type IOCAR_SR.

Attribute type: Unsigned16

Allowed values: IOCAR, IOCAR_SR, IOSAR

AR UUID

This attribute defines the UUID of the AR entry which is provided by the engineering system or the CM Initiator.

In case of companion ARs, all ARs shall use related AR UUIDs to express the strong cohesion. The relation of the AR UUIDs is indicated by a common root value; however differing in the least significant two bytes of the UUID's time_low field.

Special Case: For an AR entry used in the context of AR type IOCAR_SR, several ARs build an AR set. In this case a part of the UUID indicates the positional number of the AR within the AR-Set and the number of AR resources to allocate at this communication

endpoint. A further part of the AR UUID contains a signature over the configuration data records. The last part of the AR UUID contains a signature over the IO device's name of station.

For coding of the AR UUID see IEC 61158-6-10.

Attribute type: UUID

Session Key

This attribute shall be increased by one for each connect service by the CM Initiator and stored within the whole session.

NOTE 1 The Session Key allows the CM Initiator to detect sequence errors during the establishment and release phase of an AR.

Attribute type: Unsigned16

CM Initiator MAC Add

This attribute contains the value of the IEEE 802 MAC address of the initiator (IO controller or IO supervisor).

Attribute type: OctetString[6]

CM Responder MAC Add

This attribute contains the value of the IEEE 802 MAC address of the IO device.

Attribute type: OctetString[6]

CM Initiator Object UUID

This attribute defines the UUID of the CM Initiator.

Attribute type: UUID

AR Properties

This attribute contains the following attributes:

State

This attribute contains the initial redundancy status of the AR. The value "backup" shall only be present if the attribute AR type contains the value IOCAR_SR. The value "primary" indicates the operating AR and the value "backup" indicates the hot standby AR. The "backup" AR takes over the operation in case the PRIMARY fails. The takeover is requested by the BACKUP IO controller by means of changing the APDU data status. The BACKUP IO controller shall take this action in case of receiving a frame with a Redundancy Primary Present Flag set to "no primary present".

NOTE 2 The BACKUP and PRIMARY IO Controller may communicate over additional services to reduce the reaction time. These services are beyond the scope of this specification.

Allowed values: PRIMARY, BACKUP

Supervisor Takeover Allowed

This attribute is used to enable or disable the supervisor takeover function for all submodules used by the AR.

Allowed values: ALLOWED, NOT_ALLOWED

Parameterization Server

This attribute contains the source for parameter data.

Allowed values: EXTERNAL_PRM_SERVER, CM_INITIATOR

Device Access

This attribute contains the value for the access policy to record data objects.

Allowed values shall be according to Table 266.

Table 266 – Device Access

Value	Meaning
AR_CONTEXT	Only the submodules from the ExpectedSubmoduleBlock are accessible
DEVICE_CONTEXT	Submodule access over an IOSAR is controlled by IO device application

Companion AR

This attribute contains the value to identify companion ARs.

Allowed values shall be according to Table 267.

Table 267 – Companion AR

Value	Meaning
SINGLE_AR	RT_CLASS_1, RT_CLASS_2, RT_CLASS_3, or RT_CLASS_UDP connection establishment
FIRST_COMPANION_AR	For future use
SECOND_COMPANION_AR	For future use

In case of companion ARs the rules for companion ARUIDs shall be applied.

Acknowledge Companion AR

This attribute contains whether acknowledge for companion ARs is needed.

Allowed values shall be according to Table 268.

Table 268 – Acknowledge Companion AR

Value	Meaning
UNACKNOWLEDGED	RT_CLASS_1, RT_CLASS_2, RT_CLASS_3, or RT_CLASS_UDP connection establishment
ACKNOWLEDGED	For future use.

Startup Mode

This attribute indicates the startup mode of the application relation.

Allowed values shall be according to Table 269.

Table 269 – Startup Mode

Value	Meaning
Legacy	Recommended Compatibility mode for legacy nodes. RT_CLASS_3: The two provider mode for the RT_CLASS_3 startup is required. Compatibility mode for legacy nodes.
Advanced	RT_CLASS_3: The one provider mode for the RT_CLASS_3 startup is required. Necessary for SendClocks less than 250 µs and the usage of DFP or MRPD. Recommended for SendClocks greater or equal 250 µs.

Pull Module Alarm Allowed

This optional attribute contains whether issuing the pull module alarm is allowed for the IO device.

Allowed values shall be according to Table 270.

Table 270 – Pull Module Alarm Allowed

Value	Meaning
NOT_ALLOWED	The AlarmType(=Pull) shall signal pulling of submodule and module. The subslot number zero shall code pulling of module in conjunction with AlarmType(=Pull).
ALLOWED	The AlarmType(=Pull) shall signal pulling of submodule. The Alarm-Type(=Pull module) shall signal pulling of module. The subslot number 0 to 0x9FFF shall code pulling of submodule in conjunction with AlarmType(=Pull).

The attribute shall not be set to ALLOWED on IO devices not supporting the feature “Pull Module Alarm”. Whether an IO device supports the feature is indicated by GSD.

CM Initiator Activity Timeout Factor

This attribute defines the CM Initiator Activity Timeout Factor for the connection establishment phase. It defines the timeout value for monitoring the time period between sending the Connection response service primitive and receiving a new service request from the IO controller. The IO device application stops this monitoring when the IO CRs start their operation. Further monitoring is done by the IO CR with the values based on the CR attributes Data Hold Factor.

Attribute type: Unsigned16

Allowed value: 100 to 1 000 on startup mode “Advanced”, 1 to 1 000 on startup mode “Legacy”

Time Base: 100 ms

Remote Application Ready Timeout

This attribute defines the time the IO controller uses to monitor the time between receiving the Prm End confirmation and Application Ready indication.

Attribute type: Unsigned16

Allowed value: 300

Time Base: 1 s

Initiator UDP RT Port

This attribute defines the UDP Port which is used for RT_CLASS_UDP communication.

If no RT_CLASS_UDP is used, the value shall be set to 0x8892.

Attribute type: Unsigned16

Allowed value: 0x0400 to 0xFFFF

Responder UDP RT Port

This attribute defines the UDP Port which is used for RT_CLASS_UDP communication.

If no RT_CLASS_UDP is used, the value shall be set to 0x8892.

Attribute type: Unsigned16

Allowed value: 0x0400 to 0xFFFF

CM Initiator Station Name

This attribute defines the IP address or name of the CM Initiator.

The following syntax applies:

- 1 or more labels, separated by [.]
- total length is 1-240
- label length is 1-63
- labels consist of [a-z0-9-]
- labels do not start with [-]
- labels do not end with [-]
- the first label does not start with "port-xyz" or "port-xyz-abcde" with a,b,c,d,e, x, y, z = 0...9
- station-names do not have the form n.n.n.n, n = 0...999
- labels do only start with 'xn-' if IETF RFC 3490 is applied

Furthermore, the definition of IETF RFC 3490 shall be applied.

EXAMPLE 1 "device-1.machine-1.plant-1.vendor"

EXAMPLE 2 "device-1.bögeholz" is coded as "device-1.xn-bögeholz-90a"

NOTE 3 The field CM Initiator Station Name is not terminated by zero.

Attribute type: VisibleString[240]

List of CRs

This attribute list shall have at least one entry. It contains the following attributes:

CREP

This attribute is the local identifier for the CR entry.

Parameter Server Block

This optional attribute is reserved for future use and contains the following attributes:

Parameter Server Object UUID

This attribute defines the UUID of the Parameter Server.

Attribute type: UUID

Parameter Server Properties

This attribute defines properties of the Parameter Server.

Attribute type: Unsigned32

CM Initiator Activity Timeout Factor

This attribute defines the CM Initiator Activity Timeout Factor for the connection establishment phase between the IO device and the Parameter Server.

Attribute type: Unsigned16

Allowed value: 1 to 1 000

Time Base: 100 ms

Parameter Server Station Name

This attribute defines the IP address or name of the Parameter Server.

The syntax of the attribute CM Initiator Station Name shall be applied.

Attribute type: VisibleString[240]

AR RPC Server Block

This attribute is used to allow direct access to the RPC servers and contains the following attributes:

Initiator RPC Server Port

This attribute defines the UDP Port which is used for CM Initiator RPC server. According IANA it is recommended to use a range from 0xC000 to 0xFFFF.

Attribute type: Unsigned16

Allowed value: 0x0400 to 0xFFFF

Responder RPC Server Port

This attribute defines the UDP Port which is used for CM Responder RPC server. According IANA it is recommended to use a range from 0xC000 to 0xFFFF.

Attribute type: Unsigned16

Allowed value: 0x0400 to 0xFFFF

IR Info Block

This optional attribute shall be used by the IO device to determine whether the application relation references to IR data, which is part of the persistent PDev parameterization. The parameter shall be used only when using RT_CLASS_3 with advanced startup mode.

The parameter allows multiple IO controllers sharing the IO device using RT_CLASS_3. One of the IO controllers parameterizes the IR data at the PDev, all other IO controller reference the IR data configured by that IO controller via the IR Info Block.

The attribute consists of the following attributes:

IR Data UUID

This attribute contains the unique identifier of the isochronous realtime project which is configured by the engineering system.

Attribute type: UUID

Number of IO CRs

This parameter contains the number of IO CRs related to DFP. The parameter shall be set to "0" if no DFP is used, otherwise to the number of IO CRs related to DFP.

Attribute type: Unsigned16

List of DFP IO CRs

This list consists of the following attributes:

IO CR Reference

This parameter contains the reference to the related DFP IO CR.

Attribute type: Unsigned16

Sub Frame Offset

This parameter contains the Number of octets from the begin of the frame C_SDU to the begin of the according subframe C_SDU.

Attribute type: Unsigned16

Allowed values: 0x0000 to 0x059B

Sub Frame Data

This parameter consists of the following attributes:

Sub Frame Position

This attribute defines the position number of the sub frame.

Attribute type: Unsigned8

Allowed values: 0x01 to 0x7F

Sub Frame Data Length

This attribute defines the length of the sub frame.

Attribute type: Unsigned8

Allowed values: 0x01 to 0xFF

SR Info Block

This optional attribute defines the system redundancy parameters of the desired AR. This attribute is present only if the startup mode of the AR is set to “Advanced”. It consists of the following attributes:

Redundancy Data Hold Factor

This attribute contains the time which shall be used to hold the outputs while waiting for an application relation to become primary.

Attribute type: Unsigned16

Time base: 1 ms

Allowed value: 0x0001 to 0x00C8 (optional), 0x00C9 to 0xEA60 (mandatory), 0xEA61 to 0xFFFF (optional)

SR Properties

This attribute is used to define system redundancy behavior. It consists of the following attributes:

Input Valid on Backup AR

This attribute defines whether the IO controller awaits valid input data also on a backup AR.

Allowed values shall be according to Table 271.

Table 271 – Input Valid on Backup AR

Value	Meaning
IRRELEVANT	The device may deliver valid input data in a backup AR
REQUIRED	The device shall deliver valid input data in a backup AR

Activate Redundancy Alarm

This attribute defines whether the IO controller wants to get redundancy alarms.

Allowed values shall be according to Table 272.

Table 272 – Activate Redundancy Alarm

Value	Meaning
INACTIVE	The device shall not send the redundancy alarm
ACTIVE	The device shall send the redundancy alarm

Redundancy Info

This optional attribute defines the system redundancy info of the desired AR. This attribute is present only if the startup mode of the AR is set to “advanced”. It consists of the following attributes:

End Point 1

This attribute contains the network interface, where the AR is hosted.

Allowed values: NOT_LEFT_AND_BELOW, LEFT_OR_BELOW

End Point 2

This attribute contains the network interface, where the AR is hosted.

Allowed values: NOT_RIGHT_AND_ABOVE, RIGHT_OR_ABOVE

List of AR Vendor Blocks

This optional attribute defines the vendor block info of the desired AR. This attribute is present only if the startup mode of the AR is set to “advanced”. The AR Vendor Block allows to transport additional information related to the AR. The definition allows (in psite of its name) to transport information specified within this specification (e.g. for common profiles), profile specific information and vendor specific information.

The key is the combination of API and APStructureIdentifier. Any possible combination may exist only once in this list.

It consists of the following attributes:

API

This attribute defines to which application process the AR Vendor Block belongs to.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

AP Structure Identifier

This attribute the structure of the attributes Data Request and Data Response.

Attribute type: Unsigned32

Allowed values shall be according to Table 273 and Table 274

Table 273 – APStructureIdentifier with API := 0

Value (hexadecimal)	Meaning
0 – 0x7FFF	Vendor specific This key defines in combination with the key API the structure of Data in ARVendorBlockReq and ARVendorBlockRes.
0x8000	Extended identification rules Used for common profile “Controller to controller communication”
0x8001 – 0xFFFF	Administrative number This key defines in combination with the key API the structure of Data in ARVendorBlockReq and ARVendorBlockRes.

Table 274 – APStructureIdentifier with API != 0

Value (hexadecimal)	Meaning
0 – 0xFFFF	Administrative number for application profiles This key defines in combination with the key API the structure of Data in ARVendorBlockReq and ARVendorBlockRes.

Data Request

This attribute contains the vendor block request data

For the special case API == 0 and APStructureIdentifier == 0x8000, the request data contains the expected extended identification info as defined by the common profile “Controller to controller communication”.

Attribute type: OctetString[512]

Data Response

This attribute contains the vendor block response data

For the special case API == 0 and APStructureIdentifier == 0x8000, the request data contains the real extended identification info as defined by the common profile “Controller to controller communication”.

Attribute type: OctetString[512]

Fast Startup Mode

This optional attribute defines the connection mode for an IO AR.

Attribute type: Unsigned16

Allowed values: NORMAL_MODE, FAST_STARTUP_MODE

AR State

This attribute defines the state of the AR.

This attribute influences the Write service behavior of the data record ASE (see 7.3.1.3.5 item a). If the attribute is set to BACKUP the Write service is neglected with Error code 1 set to “backup”.

This attribute influences the behavior of the context ASE service Application Ready (see 7.3.6.3.7). The Application Ready service confirms a parameterization sequence due to application relation establishment, plug/release of a submodule, or Prm Begin. Therefore if the attribute AR State is set to BACKUP, all data records are neglected during the parameterization sequence and thus the parameterization sequence was not performed.

If the attribute AR State of the desired AR is set to BACKUP, all data records are neglected during the parameterization sequence (see 7.3.1.3.5 item b)). Thus if the AR state is set to BACKUP when the service Prm End is initiated, the Submodule State.AR Info of all submodules related to the parameterization sequence is set to the submodule's state according to the AR with AR state set to PRIMARY or if not existing set to FIRST). If submodules are added (Configuration in RUN), which are not yet contained in the PRIMARY or FIRST AR, and the submodule is existing or a decent substitute, the submodule's AR state is set to APPLICATION_READY_PENDING.

Thus if the AR state is set to BACKUP when the service Prm End is initiated, the Submodule State.AR Info of all submodules related to the parameterization sequence is set to APPLICATION_READY_PENDING, indicating that the parameterization sequence failed.

It shall be set to PRIMARY, if it is the primary AR of an AR set or if it is a “standalone”.

For the first application relation having ARType IOCARSR and belonging to a new AR Set this attribute shall be set to FIRST up to ApplicationReady. After ApplicationReady the AR state of an AR having ARType IOCARSR is set similar to the state (i.e. to PRIMARY or BACKUP) as intended by the IO controller.

For ARs not having ARType IOCARSR the attribute is set to PRIMARY always.

Attribute type: Unsigned16

Allowed values: FIRST, BACKUP, PRIMARY

Default value: PRIMARY

7.3.9.4 CRL class specification

7.3.9.4.1 Template

The CRL device object is described by the following template:

ASE:
CLASS:
CLASS ID:
PARENT CLASS:
ATTRIBUTES:

AR ASE

CRL

not used

TOP

1	(m)	Key Attribute: Implicit
2	(m)	Attribute: List of CRL Entries
2.1	(m)	Attribute: CREP
2.2	(s)	Attribute: IO CR Block
2.2.1	(m)	Constraint: IO CR type
2.2.2	(m)	Attribute: IO CR Reference
2.2.3	(m)	Attribute: LT Field
2.2.4	(m)	Attribute: IO CR Properties
2.2.4.1	(m)	Attribute: RT Class
2.2.5	(m)	Attribute: C_SDU Length
2.2.6	(m)	Attribute: Frame ID
2.2.7	(m)	Attribute: Send Clock Factor
2.2.8	(m)	Attribute: Reduction Ratio
2.2.9	(m)	Attribute: Phase
2.2.10	(m)	Attribute: Sequence
2.2.11	(m)	Attribute: Frame Send Offset
2.2.12	(m)	Attribute: Data Hold Factor
2.2.13	(m)	Attribute: IO CR Tag Header
2.2.13.1	(m)	Attribute: Vlan ID
2.2.13.2	(m)	Attribute: IO CR User Priority
2.2.14	(m)	Attribute: IO CR Multicast MAC Add
2.2.15	(m)	Attribute: List of APIs
2.2.15.1	(m)	Attribute: API
2.2.15.2	(m)	Attribute: List of Related IO Data Objects
2.2.15.2.1	(m)	Attribute: Slot Number
2.2.15.2.2	(m)	Attribute: Subslot Number
2.2.15.2.3	(m)	Attribute: IO Data Object Frame Offset
2.2.15.3	(m)	Attribute: List of Related IOCS
2.2.15.3.1	(m)	Attribute: Slot Number
2.2.15.3.2	(m)	Attribute: Subslot Number
2.2.15.3.3	(m)	Attribute: IOCS Frame Offset
2.2.16	(o)	Attribute: List of Sub Frame Data
2.2.16.1	(m)	Attribute: Sub Frame Position
2.2.16.2	(m)	Attribute: Sub Frame Length
2.2.17	(o)	Attribute: IR Frame Data
2.2.17.1	(m)	Attribute: IOCR Tx Ports
2.2.17.1.1	(m)	Attribute: Port
2.2.17.1.2	(m)	Attribute: Redundant Port
2.2.17.2	(m)	Attribute: Frame Send Offset
2.2.18	(o)	Attribute: Multicast Consumer Data
2.2.18.1	(m)	Attribute: Address Resolution Properties
2.2.18.2	(m)	Attribute: MCI Timeout Factor
2.2.18.3	(m)	Attribute: Provider Station Name

2.3	(s) Attribute:	Alarm CR Block
2.3.1	(m) Constraint:	Alarm CR type
2.3.2	(m) Attribute:	LT Field
2.3.3	(m) Attribute:	Alarm CR Properties
2.3.3.1	(m) Attribute:	Priority
2.3.3.2	(m) Attribute:	Transport
2.3.4	(m) Attribute:	RTA Timeout Factor
2.3.5	(m) Attribute:	RTA Retries
2.3.6	(m) Attribute:	Local Alarm Reference
2.3.7	(m) Attribute:	Remote Alarm Reference
2.3.8	(m) Attribute:	Max Alarm Data Length
2.3.9	(m) Attribute:	Alarm CR Tag Header High
2.3.9.1	(m) Attribute:	Vlan ID
2.3.9.2	(m) Attribute:	Alarm User Priority
2.3.10	(m) Attribute:	Alarm CR Tag Header Low
2.3.10.1	(m) Attribute:	Vlan ID
2.3.10.2	(m) Attribute:	Alarm User Priority

7.3.9.4.2 Attributes

Implicit

The key attribute Implicit indicates that the CRL object is implicitly addressed by the service.

List of CRL Entries

This attribute is composed of the following list elements:

CREP

This static attribute contains the local identifier of this CR.

IO CR Block

This selective attribute consists of the following attributes:

NOTE 1 A certain CR can either be an IOCR or an Alarm CR.

IO CR type

This attribute indicates the type of this CR.

Attribute type: Unsigned16

Allowed values: INPUT_CR, OUTPUT_CR, MULTICAST_PROVIDER_CR, MULTICAST_CONSUMER_CR

IO CR Reference

This attribute contains the identifier for the CR.

Attribute type: Unsigned16

LT Field

This attribute contains the length type field of the Physical Layer Frame.

Attribute type: Unsigned16

Allowed values: 0x8892 = IEC_61158_TYPE_10, others reserved

IO CR Properties

This attribute consists of the following attributes:

RT Class

The value RT Class defines the Frame ID range (see Table 275) to be used for communication and in consequence a traffic class (RED, ORANGE or GREEN). The traffic class is used at the provider of the frame on its injection. The traffic class is used in the bridge for forwarding by selecting the used transport

mechanism and the switching model (see 6.3.8, 6.3.10, and 6.3.11). Any bridge uses the maximum possible traffic class for the frame depending on reserved bandwidth and being in sync.

Attribute type: Unsigned8

Allowed values: 0x01 (only to be used for legacy devices), 0x02, 0x03 and 0x04

Table 275 – Traffic Classes versus RT Class

RT Class	Startup Mode	Frame ID Range	Traffic Class	Used for
0x01	Advanced, Legacy	range 7	GREEN	RT_CLASS_1 (legacy)
0x02	Advanced	range 6	GREEN	RT_CLASS_1
0x02	Legacy	range 6	ORANGE and GREEN	RT_CLASS_1, RT_CLASS_2
0x03	Advanced, Legacy	range 3	RED	RT_CLASS_3
0x02	Advanced, Legacy	range 7	GREEN	RT_CLASS_UDP

For the values of the particular frame ID ranges see Table 276.

In traffic class GREEN IEEE 802.3, IEEE 802.1D, IEEE 802.1Q, and IP-Suite (see 6.3.12) are applicable without any extension.

In traffic class ORANGE IEEE 802.3, IEEE 802.1D are applicable without any extension. Extensions of IEEE 802.1Q are specified in 6.3.10.2. The traffic class ORANGE exists only in conjunction with AR Properties.Startup Mode = Legacy (see 7.3.9.3) and if an ORANGE period is defined (see 6.3.9.2.3).

In traffic class RED IEEE 802.3, IEEE 802.1D and IEEE 802.1Q are applicable with extensions specified in 6.3.8, 6.3.10, and 6.3.11.

C_SDU Length

This attribute contains the overall length of the transmitted Data (CSDU or else CSFSDU Length); thus for a dynamically packed frame it indicates the length of all sub frames including the sub frame this IO device contributes.

Attribute type: Unsigned16

Allowed Values for RT_CLASS_1, RT_CLASS_2 and RT_CLASS_3: 40 to 1 440

NOTE 2 The lower bound of the C_SDU length is derived from the minimum frame size of an Ethernet frame (= 64 bytes). On subtracting the size of the header (i.e. destination address, source address and type) and the size of the footer (i.e. CRC) 46 bytes are left. On subtracting the size of the RT frame header (i.e. Frame ID) and footer (i.e. cycle counter, data status and transfer status) the minimum length of the C_SDU is 40 bytes.

Even if a VLAN tag is used (sized 4 bytes), the minimum C_SDU length stays on 40 bytes, since the VLAN tag may be stripped by some switches, leaving still an Ethernet frame with a minimum of 64 bytes.

Allowed Values for RT_CLASS_UDP: 12 to 1 440

NOTE 3 The lower bound of the C_SDU length is derived from the minimum frame size of an Ethernet frame (= 64 bytes). On subtracting the size of the header (i.e. destination address, source address and type) and the size of the footer (i.e. CRC) 46 bytes are left. From these 46 bytes the IP header (20 bytes) and the UDP header (8 bytes) are subtracted, leaving 18 bytes. On subtracting the size of the RT frame header (i.e. Frame ID) and footer (i.e. cycle counter, data status and transfer status) the minimum length of the C_SDU is 12 bytes.

Again the VLAN tag is not included in the computation of the minimum C_SDU length.

Frame ID

This attribute contains the identifier of the data within the frame. The value of the frame ID is provided by either the consumer of the frame or else by the engineering system; see Table 276 for details.

Attribute type: Unsigned16

The allowed values according to Table 276 shall be used.

Table 276 – Frame ID

Range	Value (hexadecimal)	Meaning	Use
range 3	0x0100 – 0x06FF (provided by ES)	Dedicated to communication class RT_CLASS_3 (RED) unicast and multicast	non redundant, normal or DFP
	0x0700 – 0x0FFF (provided by ES)	Dedicated to communication class RT_CLASS_3 (RED) unicast and multicast	redundant, normal or DFP
range 6	0x8000 – 0xBBFF (provided by consumer)	Dedicated to communication class RT_CLASS_2 (ORANGE, legacy) and RT_CLASS_1 (GREEN) unicast	non redundant, normal
	0xBC00 – 0xBFFF (provided by ES)	Dedicated to communication class RT_CLASS_2 (ORANGE, legacy) and RT_CLASS_1 (GREEN) multicast	non redundant, normal
range 7	0xC000 – 0xF7FF (provided by consumer)	Dedicated to communication class RT_CLASS_UDP (GREEN) and RT_CLASS_1 (GREEN, legacy) unicast	non redundant, normal Disjunct Frame IDs shall be used if concurrent usage in different ARs
	0xF800 – 0xFBFF (provided by ES)	Dedicated to communication class RT_CLASS_UDP (GREEN) and RT_CLASS_1 (GREEN, legacy) multicast	non redundant, normal Disjunct Frame IDs shall be used if concurrent usage in different ARs
	Other	Reserved	Shall not be used in IO CR

The column “Use” indicates:

redundant (meaning media redundancy): In this case two FrameIDs (e.g. 0xFFFF0 and 0xFFFF1) are used for one CR; one for each direction in which the frame travels through the ring. The pair shall be identified ignoring the least significant bit of the FrameID.

non redundant (meaning media redundancy): One Frame ID is used.

normal: Indicates that the frame has no sub frame structure.

DFP: Indicates that the frame is dynamically packed and thus has a sub frame structure.

Send Clock Factor

This attribute contains the Send Clock Factor. The Send Clock Factor shall be set in relation to the signaling rate.

Attribute type: Unsigned16

Time Base: 31,25 µs

Allowed Values: 5 to 128 for 100 Mbps, 1 to 128 for 1 Gbps

Default Value: 32

Reduction Ratio

This attribute contains the reduction ratio of the send clock of the device. The actual send cycle of the data set is Reduction Ratio multiplied with the send clock. It defines a macro cycle for this particular CR.

Attribute type: Unsigned16

Allowed Values: see Table 277

Table 277 – Reduction Ratios

Send Clock Factor	RT_CLASS_1 RT_CLASS_2	RT_CLASS_3	RT_CLASS_UDP
1 - 32	1, 2, 4, 8, 16, 32, 64, 128, 256, 512	1, 2, 4, 8, 16 optional: 32, 64, 128, 256, 512	128, 256, 512, 1024, 2048, 4096, 8192, 16384 optional: 1, 2, 4, 8, 16, 32, 64
33 - 64	1, 2, 4, 8, 16, 32, 64, 128, 256	1, 2, 4, 8, 16 optional: 32, 64, 128, 256	128, 256, 512, 1024, 2048, 4096, 8192 optional: 1, 2, 4, 8, 16, 32, 64
65 - 128	1, 2, 4, 8, 16, 32, 64, 128	1, 2, 4, 8, 16 optional: 32, 64, 128	128, 256, 512, 1024, 2048, 4096 optional: 1, 2, 4, 8, 16, 32, 64

Phase

This attribute indicates in which particular send cycle the data has to be sent. This parameter depends from the parameter Reduction Ratio.

Attribute type: Unsigned16

Allowed values: 1 to selected Reduction Ratio.

NOTE 4 The parameter phase allows spreading the sending of the frames over the whole macro cycle and provides a means to manage distribution of the used bandwidth. The phases e.g. 1 to 4 are possible if the Reduction Ratio is set to 4.

Sequence

This attribute contains the position of the data frame in the sending queue. The value 0 shall be used to indicate an undefined sequence.

Attribute type: Unsigned16

Frame Send Offset

This attribute contains the relative send offset to the start of the related reduction ratio and phase. The time base is 1 ns. The value shall be less than the duration of one phase.

Attribute type: Unsigned32

Allowed Values: BEST_EFFORT – (mandatory) as soon as possible, 0 – 0x003D08FF for optimized scheduling

Data Hold Factor

This attribute contains a factor, which is used to calculate the Data Hold Time. This time is the time interval for holding last output data if no IO data have been received. The interval for the Data Hold Time is the product of the attribute values of Data Hold Factor, Send Clock Factor, Reduction Ratio and the time base of 31,25 µs related to the monitored consumer. The Data Hold Time shall equal or less 1,92 s if the value of the AR attribute RT Class is 0x01, 0x02 or 0x03 and equal or less 49,152 s if the value of the AR attribute RT Class is 0x04.

Attribute type: Unsigned16

Allowed values: 0x0001 to 0x1E00

IO CR Tag Header

This attribute consists of the following attributes:

Vlan ID

This attribute contains the IEEE 802.1Q VLAN ID. The default value is NO_VLANID that indicates that the tag header contains only priority information.

Allowed values: according to IEEE 802.1Q

IO CR User Priority

This attribute contains the IEEE 802.1Q VLAN ID priority. The default value is IOCR_PRIORITY.

Allowed values: IOCR_PRIORITY – higher priority than IP traffic

IO CR Multicast MAC Add

This attribute contains the multicast address according IEEE 802.3 in case of multicast transmission. The value shall be set to zero for unicast connections.

Attribute type: OctetString[6]

List of APIs

This attribute list shall contain at least one API that the CR belongs to. This list describes where the IO Data Objects and IOCS' of the individual submodules indicated by API, Slot and Subslot Number are placed into the frame. Thus this attribute describes the frame layout related to this IO device.

For a dynamically packed frame only the layout of the sub frame related to this IO device is described.

A list element consists of the following attributes:

API

This attribute defines to which application process the IOCR belongs to.

Attribute type: Unsigned32

Allowed values: 0 to 0xFFFFFFFF

List of Related IO Data Objects

This attribute contains the IO Data Objects that are related to the CR. A List element consists of the following attributes:

Slot Number

This attribute defines to which slot the IO Data belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Subslot Number

This attribute defines to which subslot the IO Data belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

IO Data Object Frame Offset

This attribute defines the offset of the DataItem. The sum of IO Data Object Frame Offset, Submodule Data Length and Length IOPS shall not exceed the C_SDU Length.

Attribute type: Unsigned16

Allowed values: 0 to 1 439

List of Related IOCS

This attribute contains the IOCS that are related to the CR. A list element consists of the following attributes:

Slot Number

This attribute defines to which slot the IOCS belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x7FFF

Subslot Number

This attribute defines to which subslot the IOCS belongs.

Attribute type: Unsigned16

Allowed values: 0 to 0x9FFF

IOCS Frame Offset

This attribute defines the offset of the IOCS of the DataItem. The sum of IOCS Frame Offset and Length IOCS shall not exceed the C-SDU Length.

Attribute type: Unsigned16

Allowed values: 0 to 1 439

List of Sub frame Data

This attribute shall be present if dynamic frame packing is used. The attribute describes the sub-structure of the frame. Dynamic frame packing is indicated by using a Frame ID within the DFP range (see the attribute Frame ID above).

The sub frames described in the list are packed one after another without gap; thus the following formula holds:

$$SubFrameOffset_n = 2 + \sum_{i=0}^{n-1} SubFrameData_i, SubFrameDataLength \quad (23)$$

where

SubFrameOffset is the sub frame offset

SubFrameData the sub frame data

SubFrameDataLength is the length of the sub frame

All IO Data Objects and IOCS defined within the List of APIs shall be located within exactly one sub frame. The following sub frame local offsets shall not be used for IO Data Objects or IOCS to keep the sub frame header and footer intact: 0, 1, 2, 3, Sub frame Data Length - 2, Sub frame Data Length - 1.

An IO device may derive its own Sub Frame Position by traversing the list and matching its IO Data Object and IOCS Frame Offsets whether being within SubframeOffset and SubframeOffset + Subframe Data Length. If being so, the Sub frame Position is its own.

A list element consists of the following attributes:

Sub Frame Position

This attribute defines the position number of the sub frame. The position number shall be strictly increasing within the list of Sub frame Data.

Attribute type: Unsigned8

Allowed values: 0x01 to 0x3F for packetizing and unpacking of sub frames during transport and end to end; 0x40 to 0x7F for packetizing and unpacking of sub frames end to end; 0x00 and 0x80 to 0xFF reserved

Sub Frame Data Length

This attribute defines the length of the sub frame.

Attribute type: Unsigned8

Allowed values: 0x01 to 0xFF

IR Frame Data

This optional attribute shall be present, if the attribute RT Class is set to 0x03 and if the GSD of the IO device indicates that it needs IR frame data. The attribute shall not be present, if the attribute RT Class is set to another value than 0x03 or if GSD indicates that the IO device does not support IR frame data.

It consists of the following attributes:

IOCR Tx Ports

This attribute consists of

Port

This attribute indicates the port, where the frame shall be sent.

Attribute type: Unsigned8

Allowed values: 0x01 to 0xFF

Redundant Port

This attribute indicates the port, where the redundant frame shall be sent if the Media Redundancy with planned duplication is used.

This attribute shall be set to zero if not used, otherwise this attribute shall be set to the number of the port, where the redundant frame shall be sent

Attribute type: Unsigned8

Allowed values: 0x00 (mandatory), 0x01 to 0xFF (optional)

Frame Send Offset

This attribute contains the relative send offset to the start of the related reduction ratio and phase of the redundant frame. The time base is 1 ns. The value shall be less than the duration of one phase.

The value is only valid, if the Media Redundancy with planned duplication is used.

Attribute type: Unsigned32

Allowed Values: 0 to 0x003D08FF

Multicast Consumer Data

This attribute shall be present if the attribute IO CR type contains the value MULTICAST_CONSUMER_CR and consists of the following attributes:

Address Resolution Properties

This attribute defines the protocol and interval for address resolution of the multicast provider. It includes the waiting time between two subsequent address resolution cycles.

Attribute type: Unsigned32

Allowed values for protocol: DNS, DCP

Allowed values for interval: 0x0001 to 0xFFFF

Time base: 100 ms

MCI Timeout Factor

This attribute defines the monitoring interval for address resolution and in data state of the multicast provider. The application ready shall be delayed until the in data state is reached or the Multicast Initiator Timeout is reached.

Attribute type: Unsigned16

Allowed values: 0x0000 to 0x0064

Time base: 100 ms

Provider Station Name

This attribute defines the IP address or name of the Multicast Provider.

The syntax of the attribute CM Initiator Station Name shall be applied.

Attribute type: VisibleString[240]

Alarm CR Block

This selective attribute consists of the following attributes:

NOTE 5 A certain CR can either be an IOCR or an Alarm CR.

Alarm CR type

This attribute indicates the type of this CR.

Attribute type: Unsigned16

Allowed value: ALARM_CR

LT Field

This attribute contains the length type field of the Physical Layer Frame.

Attribute type: Unsigned16

Allowed values: 0x8892 = IEC_61158_TYPE_10

Alarm CR Properties

This attribute consists of the following attributes:

Attribute type: Unsigned16

Priority

This attribute contains the priority of the alarm.

Allowed Values: USER_PRIORITY (default) - the priority given by the user is used and two alarm resources are available,
FIXED_PRIORITY - user priority is ignored and only one alarm resource is available

Transport

This attribute contains the transport layer of the alarm.

RTA_CLASS_1 means, IEEE 802.3, IEEE 802.1D and IEEE 802.1Q are applicable without any extension.

RTA_CLASS_UDP means, IEEE 802.3, IEEE 802.1D, IEEE 802.1Q, and IP-Suite (see 6.3.12) are applicable without any extension.

Attribute type: Unsigned8

Allowed values: RTA_CLASS_1, RTA_CLASS_UDP

RTA Timeout Factor

This attribute indicates the timeout value of the alarm source with a base of 100 ms.

Attribute type: Unsigned16

Time Base: 100 ms

Allowed Values: 1 up to 100 mandatory, 101 to $2^{16}-1$ optional

Default value: 100 ms

RTA Retries

This attribute contains the number of retries.

Attribute type: Unsigned16

Allowed Values: 3 to 15.

Local Alarm Reference

This attribute contains the identifier of the CM Initiator (e.g. IO controller).

Attribute type: Unsigned16

Remote Alarm Reference

This attribute contains the identifier of the CM Responder (e.g. IO device).

Attribute type: Unsigned16

Max Alarm Data Length

This attribute contains the maximum length of alarm data accepted by the alarm sink (IO device, IO controller, and IO supervisor).

Attribute type: Unsigned16

Allowed value: 200 to 1 432

Alarm CR Tag Header High

This attribute consists of the following attributes:

Vlan ID

This attribute contains the IEEE 802.1Q VLAN ID. The default value is NO_VLAN that indicates that the tag header contains only priority information.

Allowed values: according to IEEE 802.1Q

Alarm User Priority

This attribute contains the priority of the alarm.

Allowed Values: USER_PRIORITY_HIGH

Alarm CR Tag Header Low

This attribute consists of the following attributes:

Vlan ID

This attribute contains the IEEE 802.1Q VLAN ID. The default value is NO_VLAN that indicates that the tag header contains only priority information.

Allowed values: according to IEEE 802.1Q

Alarm User Priority

This attribute contains the priority of the alarm.

Allowed Values: USER_PRIORITY_LOW

7.3.9.4.3 Invocation of CRL object

For the invocation of the CRL object the following rule has to be considered:

- Only one CRL object shall be invoked in a IO device.

7.3.9.5 AR service specification

7.3.9.5.1 General Data Definition

All services to read the AR Data object have the same parameter layout. To ease the description the parameter is thus described once upfront to the services. The services itself act as a filter for the information from the AR ASE.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1); for the error code hierarchy of the record data object 7.3.1.3.5 applies.

The parameter Data has a service specific structure and meaning. Table 278 shows the specific elements of Data for this service. Note that dependent on the particular service described below only parts of the information are returned.

Table 278 – Data elements of Read AR Data

Parameter name	service specific elements	rsp	cnf
Data	List of ARs	M	M(=)
	AR UUID	M	M(=)
	CM Initiator Object UUID	M	M(=)
	Parameter Server Object UUID	M	M(=)
	AR Properties	M	M(=)
	AR type	M	M(=)
	Alarm CR type	M	M(=)
	Local Alarm Reference	M	M(=)
	Remote Alarm Reference	M	M(=)
	Initiator UDP RT Port	M	M(=)
	Responder UDP RT Port	M	M(=)
	CM Initiator Station Name Length	M	M(=)
	CM Initiator Station Name	M	M(=)
	Parameter Server Station Name Length	M	M(=)
	Parameter Server Station Name	C	C(=)
	Number of IOCRs	M	M(=)
	List of IOCRs	M	M(=)
	IO CR Properties	M	M(=)
	IO CR type	M	M(=)
	Frame ID	M	M(=)
	APDU Status	M	M(=)
	Number of APIs	M	M(=)
	List of APIs	M	M(=)
	API	M	M(=)
	AR Data Info	M	M(=)
	Number Of Entries	M	M(=)
	IR Info Block	C	C(=)

Parameter name	service specific elements	rsp	cnf
	IR Data UUID	M	M(=)
	Number of IO CRs	M	M(=)
	List of Sub Frame Data	C	C(=)
	Frame ID	M	M(=)
	Sub Frame Offset	M	M(=)
	Sub Frame Data	M	M(=)
	SR Info Block	C	C(=)
	Redundancy Data Hold Factor	M	M(=)
	SR Properties	M	M(=)
	Input Valid on Backup AR	M	M(=)
	Activate Redundancy Alarm	M	M(=)
	List of AR Vendor Block Requests	C	C(=)
	AP Structure Identifier	M	M(=)
	API	M	M(=)
	Data Request	M	M(=)
	List of AR Vendor Block Responses	C	C(=)
	AP Structure Identifier	M	M(=)
	API	M	M(=)
	Data Response	M	M(=)
	AR FSU Block	C	C(=)
	FS Parameter Block	C	CU(=)
	Mode	M	M(=)
	FS Parameter UUID	M	M(=)
	Fast StartUp Block	C	C(=)
	Data	M	M(=)
	SRL Data	C	C(=)
	Redundancy Info	M	M(=)
	End Point 1	M	M(=)
	End Point 2	M	M(=)

Data

This parameter contains the values of the object which has been read.

List of ARs

This parameter is composed of the following list elements:

AR UUID

This parameter contains the value of the corresponding attribute of the ASE object.

CM Initiator Object UUID

This parameter contains the value of the corresponding attribute of the Context ASE object.

Parameter Server Object UUID

This parameter contains the value of the corresponding attribute of the ASE object.

AR Properties

This parameter contains the value of the corresponding attribute of the ASE object.

AR type

This parameter contains the value of the corresponding attribute of the ASE object.

Alarm CR type

This parameter contains the value of the corresponding attribute of the ASE object.

Local Alarm Reference

This parameter contains the value of the corresponding attribute of the ASE object.

Remote Alarm Reference

This parameter contains the value of the corresponding attribute of the ASE object.

Initiator UDP RT Port

This parameter contains the value of the corresponding attribute of the ASE object.

Responder UDP RT Port

This parameter contains the value of the corresponding attribute of the ASE object.

CM Initiator Station Name Length

This parameter contains the value of the corresponding attribute of the ASE object.

CM Initiator Station Name

This parameter contains the value of the corresponding attribute of the ASE object.

Parameter Server Station Name Length

This parameter contains the value of the corresponding attribute of the ASE object.

Parameter Server Station Name

This parameter contains the value of the corresponding attribute of the ASE object.

Number of IOCRs

This parameter contains the value of the corresponding attribute of the ASE object.

List of IOCRs

This parameter list is composed of the following list elements:

IO CR Properties

This parameter contains the value of the corresponding attribute of the ASE object.

IO CR type

This parameter contains the value of the corresponding attribute of the ASE object.

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

APDU Status

This parameter contains the value of the corresponding attribute of the ASE object.

List of APIs

This parameter list is composed of the following list elements:

API

This parameter contains the value of the corresponding attribute of the ASE object.

AR Data Info

This parameter is composed of the following elements

Number Of Entries

This parameter contains the number of blocks in the AR Data Info.

Attribute type: Unsigned16

IR Info Block

This optional parameter contains the value of the corresponding attribute of the ASE object.

IR Data UUID

This parameter contains the value of the corresponding attribute of the ASE object.

Number of IO CRs

This parameter contains the value of the corresponding attribute of the ASE object.

List of Sub Frame Data

This parameter contains the value of the corresponding attribute of the ASE object.

Frame ID

This parameter contains the value of the corresponding attribute of the ASE object.

Sub Frame Offset

This parameter contains the value of the corresponding attribute of the ASE object.

Sub Frame Data

This parameter contains the value of the corresponding attribute of the ASE object.

SR Info Block

This optional parameter contains the value of the corresponding attribute of the ASE object.

Redundancy Data Hold Factor

This parameter contains the value of the corresponding attribute of the ASE object.

SR Properties

This parameter contains the value of the corresponding attribute of the ASE object.

Input Valid on Backup AR

This parameter contains the value of the corresponding attribute of the ASE object.

Activate Redundancy Alarm

This parameter contains the value of the corresponding attribute of the ASE object.

List of AR Vendor Block Requests

This optional parameter contains the value of the corresponding attribute of the ASE object.

AP Structure Identifier

This parameter contains the value of the corresponding attribute of the ASE object.

API

This parameter contains the value of the corresponding attribute of the ASE object.

Data Request

This parameter contains the value of the corresponding attribute of the ASE object.

List of AR Vendor Block Responses

This optional parameter contains the value of the corresponding attribute of the ASE object.

AP Structure Identifier

This parameter contains the value of the corresponding attribute of the ASE object.

API

This parameter contains the value of the corresponding attribute of the ASE object.

Data Response

This parameter contains the value of the corresponding attribute of the ASE object.

AR FSU Block

This optional parameter contains the value of the corresponding attribute of the ASE object.

FS Parameter Block

This optional parameter consists of the following elements:

FS Parameter Mode

This parameter consists of the following elements:

Mode

This parameter contains the value of the corresponding attribute of the ASE object.

FS Parameter UUID

This parameter contains the value of the corresponding attribute of the ASE object.

Fast Startup Block

This optional parameter consists of the following elements:

Data

This parameter contains the value of the corresponding attribute of the ASE object.

SRL Data

This optional parameter contains the value of the corresponding attribute of the ASE object.

Redundancy Info

This parameter consists of the following elements:

End Point 1

This parameter contains whether the left/below node is the delivering node.

Attribute type: Boolean

End Point 2

This parameter contains whether the right/above node is the delivering node.

Attribute type: Boolean

7.3.9.5.2 Read AR Data for one API

This confirmed service may be used to read the AR Data for a particular API. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 278 shows the specific elements of Data for this service.

AR Data for one API belongs to an AP, which is addressed by its API. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service filters the AR ASE information regarding to the addressed API, and returns information for this API.

7.3.9.5.3 Read AR Data for one device

This confirmed service may be used to read the AR Data for a particular API. This service shall only be used in conjunction with the implicit AR, IO AR or Supervisor AR.

The parameter structure and the meaning of the parameters is the same as described in the read service (see 7.3.1.4.1.1). But the parameter Data has a service specific structure and meaning. Table 278 shows the specific elements of Data for this service.

AR Data for one device belongs to the device. The parameter Index identifies the data record index as specified in IEC 61158-6-10.

This service returns the AR ASE information of this device.

7.3.9.5.4 Local Set AR State

This service shall be used to set the state of the addressed AR. Table 279 shows the parameters of the service.

Table 279 – Local Set AR State

Parameter name	req	cnf
Argument	M(=)	M(=)
AREP	M(=)	M(=)
AR State	M(=)	

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

AR State

This parameter contains the value of the corresponding attribute of the ASE object.

7.3.9.5.5 Local AR In Data

This service indicated that the AR is in state “IN DATA”. The next New Data_ind () after the “Local AR In Data.ind” contains the first valid data from the IO controller.

Table 280 shows the parameters of the service.

Table 280 – Local AR In Data

Parameter name	ind
Argument	M(=)
AREP	M(=)

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR.

7.4 Behavior of an IO device

7.4.1 Overview

Each IO Device shall have the following ASE state machines:

- Startup SM
- Submodule SM
- Alarm Queue SM

NOTE Actually, the Alarm Queue SM is just an instance storing alarms in order to bring them in a coordinated sequence rather than a real state machine.

Each IO Device supporting Isochronous Mode ASE shall additionally have the following ASE state machine:

- Input SM
- Output SM
- SyncCtl SM

Figure 44 shows the overview of the ASE machines.

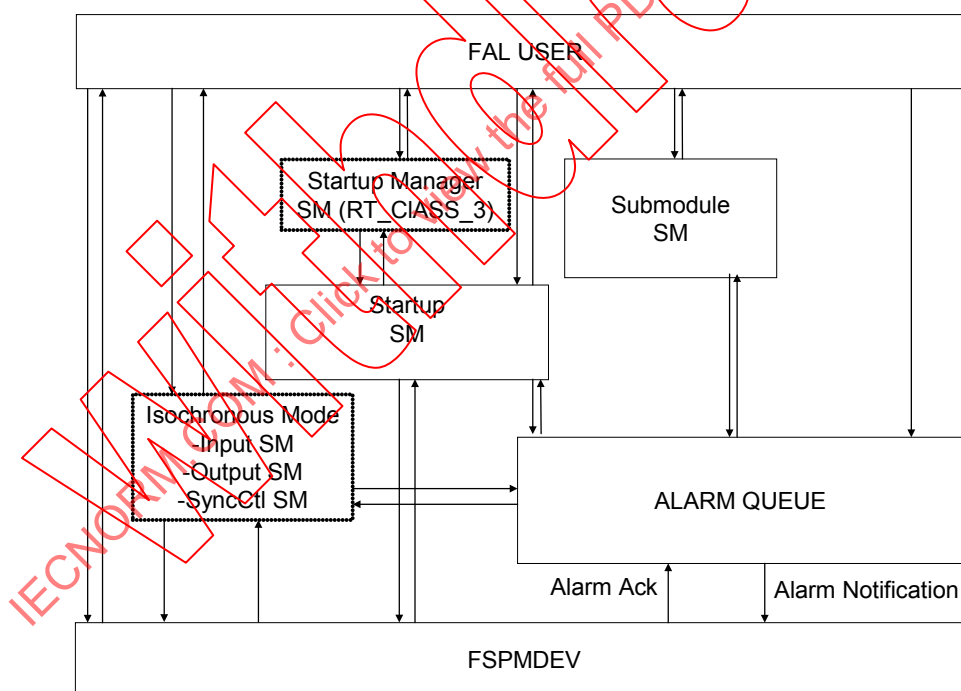


Figure 44 – Overview ASE state machines for IO device

7.4.2 Startup of an IO device

7.4.2.1 Common

This subclause describes the application behavior of an IO device during startup of the user application.

It includes extracts of the behavior of the following ASEs:

- Context ASE
- Record Data ASE
- IO Data ASE

NOTE This subclause provides selected behavior only and is not a comprehensive definition of the object behavior of the above mentioned ASEs.

7.4.2.2 Primitive definitions

7.4.2.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by an IO device during startup state machine (DEVSM) are described in the service definition and shown in Table 281.

Table 281 – Remote primitives issued or received by DEVSM

Primitive	Source	Destination	Associated parameters	Functions
Alarm.cnf (+)	FAL	DEVSM	see 7.3.5.3	—
Application Ready.cnf (+)	FAL	DEVSM	see 7.3.6.3.7	—
Connect.ind	FAL	DEVSM	see 7.3.6.3.1.2	—
Prm End.ind	FAL	DEVSM	see 7.3.6.3.6	—
Read Input Data.ind	FAL	DEVSM	see 7.3.2.5.9	—
Read Output Data.ind	FAL	DEVSM	see 7.3.2.5.16	—
Write.ind	FAL	DEVSM	see 7.3.1.4.2.1	—
Alarm.req	DEVSM	FAL	see 7.3.5.3	—
Application Ready.req	DEVSM	FAL	see 7.3.6.3.7	—
Connect.rsp (+)	DEVSM	FAL	see 7.3.6.3.1.2	—
Prm End.rsp (+)	DEVSM	FAL	see 7.3.6.3.6	—
Read Input Data.rsp (-)	DEVSM	FAL	see 7.3.2.5.9	—
Read Input Data.rsp (+)	DEVSM	FAL	see 7.3.2.5.9	—
Read Output Data.rsp (-)	DEVSM	FAL	see 7.3.2.5.16	—
Read Output Data.rsp (+)	DEVSM	FAL	see 7.3.2.5.16	—
Write.rsp (-)	DEVSM	FAL	see 7.3.1.4.2.1	—
Write.rsp (+)	DEVSM	FAL	see 7.3.1.4.2.1	—

7.4.2.2.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by DEVSM are described in the service definition and shown in Table 282.

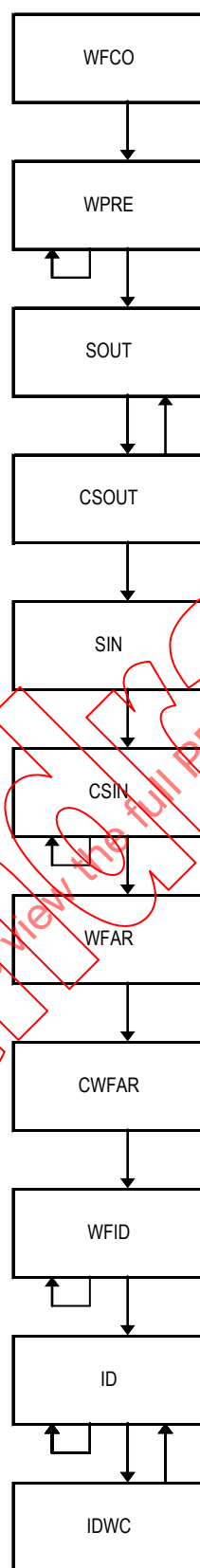
Table 282 – Local primitives issued or received by DEVSM

Primitive	Source	Destination	Associated parameters	Functions
Local Get Output_cnf (+)	FAL	DEVSM	see 7.3.2.5.12	—
Local Get Output_req	DEVSM	FAL	see 7.3.2.5.12	—
Local New Output_ind	FAL	DEVSM	see 7.3.2.5.14	—
Local Set Input_cnf (+)	FAL	DEVSM	see 7.3.2.5.1	—
Local Set Input_req	DEVSM	FAL	see 7.3.2.5.1	—
Local Set Output IOCS_cnf (+)	FAL	DEVSM	see 7.3.2.5.11	—
Local Set Output IOCS_req	DEVSM	FAL	see 7.3.2.5.11	—

7.4.2.3 State transition diagram

Figure 45 shows the state diagram DEVSM.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-10:2014

**Figure 45 – State transition diagram DEVSM****States of the DEVSM****WFCO**

Wait for a Connect indication

WPRE	Wait for a Prm End indication
SOUT	Initialize output according parameterization
CSOUT	Wait for output initialization
SIN	Initialize input buffer according to the real inputs
CSIN	Wait for input buffer initialization
WFAR	Wait for user application ready
CWFAR	Wait for Application Ready response
WFID	Wait for IN Data
ID	Working state IN Data
IDWC	Wait for input buffer or output update

7.4.2.4 State machine description

This state machine describes the application behavior of an IO device during startup of the user application.

7.4.2.5 State table

Table 283 defines the state table DEVSM.

Table 283 – State table DEVSM

#	Current State	Event /Condition => Action	Next State
1	WFCO	Connect.ind () /CheckExpectedCnf()==Ok => SetConnectResponseData("Zero") StoredSessionKey:=AR Parameter Block.SessionKey MCR_IOPS_BAD := FALSE Connect.rsp (+)	WPRE
2	WFCO	Connect.ind () /CheckExpectedCnf()!=Ok => DeltaConfiguration=AdaptConfiguration() StoredSessionKey:=AR Parameter Block.SessionKey SetConnectResponseData(DeltaConfiguration) MCR_IOPS_BAD := FALSE Connect.rsp (+)	WPRE
3	WPRE	Write.ind () /CheckParameter()==Ok => StoreParameter() Write.rsp (+)	WPRE
4	WPRE	Write.ind () /CheckParameter()!=Ok => SetWrErrorResponseData("ErrorCode=PNIORW", OTHER=UserSpecific) Write.rsp (-)	WPRE
5	WPRE	Prm End.ind () => Prm End.rsp (+)	SOUT

#	Current State	Event /Condition =>Action	Next State
6	SOUT	/Submodule Properties.Safe State Behavior == REPLACEMENT_VALUE => For all involved slot/subslots do: Slot.Subslot.IOPS := BAD_BY_DEVICE Output Data object subelement value:=Slot.Subslot.Safe State Value Slot.Subslot.IOCS:=GOOD For all involved slot/subslots do: Local Set Output IOCS_req ()	CSOUT
7	SOUT	/Submodule Properties.Safe State Behavior == LAST_VALUE => For all involved slot/subslots do: Slot.Subslot.IOPS := BAD_BY_DEVICE Output Data object subelement value:="local initial value" Slot.Subslot.IOCS:=GOOD For all involved slot/subslots do: Local Set Output IOCS_req ()	CSOUT
8	SOUT	/Submodule Properties.Safe State Behavior == ZERO => For all involved slot/subslots do: Slot.Subslot.IOPS := BAD_BY_DEVICE Output Data object subelement value:=ZERO Slot.Subslot.IOCS:=GOOD For all involved slot/subslots do: Local Set Output IOCS_req ()	CSOUT
9	CSOUT	Local Set Output IOCS_cnf (+) /not last one =>	SOUT
10	CSOUT	Local Set Output IOCS_cnf (+) /last one =>	SIN
11	CSOUT	Local Set Output IOCS_cnf (+) => ignore (M-Consumer submodule)	SIN
12	SIN	=> For all involved slot/subslots with submodul status GOOD do: Slot := Slot Number; Subslot := Subslot Number IOPS := GOOD Subslot Input Data:=Input Data object subelement value local IOCS:=GOOD For all involved slot/subslots with submodul status BAD do: Slot := Slot Number; Subslot := Subslot Number IOPS := BAD_BY_DEVICE local IOCS:=BAD_BY_DEVICE For all involved slot/subslots do: Local Set Input_req ()	CSIN
13	CSIN	Local Set Input_cnf (+) /not last one => ignore	CSIN
14	CSIN	Local Set Input_cnf (+) /last one =>	WFAR
15	WFAR	UserApplReady () => SessionKey:=StoredSessionKey MCR_IOPS_BAD := FALSE Application Ready.req ()	CWFAR
16	CWFAR	Application Ready.cnf (+) =>	WFID

#	Current State	Event /Condition =>Action	Next State
17	WFID	Local New Output_ind () /not last one => ignore	WFID
18	WFID	Local New Output_ind () /last one (exclude CREPs with Multicast Communication) => SumARinData ()	ID
19	ID	Local New Output_ind () => Local Get Output_req ()	IDWC
20	ID	Local New Output_ind () => "real output values are not modified, possible actions are local matter or for future use with redundancy"	ID
21	ID	Local New Output_ind () /(CREP with Multicast Communication) && (CHECK_IOPS(CREP) == FALSE) && (MCR_IOPS_BAD == TRUE) => ignore	ID
22	ID	Local New Output_ind () /(CREP with Multicast Communication) && (CHECK_IOPS(CREP) == FALSE) && (MCR_IOPS_BAD == FALSE) => MCR_IOPS_BAD := TRUE Alarm Type:=Multicast Communication Mismatch Alarm User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Appears ChannelErrorType := Multicast Provider Status Mismatch ExtChannelErrorType := Provider state changed to value bad Alarm.req ()	ID
23	ID	Local New Output_ind () /(CREP with Multicast Communication) && (CHECK_IOPS(CREP) == TRUE) && (MCR_IOPS_BAD == TRUE) => MCR_IOPS_BAD := FALSE Alarm Type:=Multicast Communication Mismatch Alarm User Structure Identifier := 0x8000 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := Multicast Provider Status Mismatch Alarm.req ()	ID
24	ID	Local New Output_ind () /(CREP with Multicast Communication) && (CHECK_IOPS(CREP) == TRUE) && (MCR_IOPS_BAD == FALSE) => Local Get Output.req ()	IDWC
25	IDWC	Local Get Output_cnf (+) /IOPS==GOOD => Slot.Subslot.IOPS := GOOD Output Data object subelement value:=Subslot Output Data	ID
26	IDWC	Local Get Output_cnf (+) /IOPS==BAD_BY_CONTROLLER & Submodule Properties.Safe State Behavior == REPLACEMENT_VALUE => Slot.Subslot.IOPS := BAD_BY_CONTROLLER Output Data object subelement value:=Slot.Subslot.Safe State Value	ID

#	Current State	Event /Condition =>Action	Next State
27	IDWC	Local Get Output_cnf (+) /IOPS==BAD_BY_CONTROLLER & Submodule Properties.Safe State Behavior == LAST_VALUE => Slot.Subslot.IOPS := BAD_BY_CONTROLLER	ID
28	IDWC	Local Get Output_cnf (+) /IOPS==BAD_BY_CONTROLLER & Submodule Properties.Safe State Behavior == ZERO => Slot.Subslot.IOPS := BAD_BY_CONTROLLER Output Data object subelement value:=ZERO	ID
29	IDWC	Local Get Output_cnf (+) /IOPS==BAD_BY_PROVIDER & Submodule Properties.Safe State Behavior == REPLACEMENT_VALUE/LAST_VALUE/ZERO => Slot.Subslot.IOPS := BAD_BY_PROVIDER Output Data object subelement value:=REPLACEMENT_VALUE/LAST_VALUE/ZERO MCR_IOPS_BAD := TRUE Alarm Type:=Multicast Communication Mismatch Alarm User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Appears ChannelErrorType := Multicast Provider Status Mismatch ExtChannelErrorType := Provider state changed to value bad Alarm.req ()	ID
30	IDWC	Local Get Output_cnf (+) /IOPS==GOOD && MCR_IOPS_BAD == TRUE => Slot.Subslot.IOPS := GOOD Output Data object subelement value:=Output Data MCR_IOPS_BAD := FALSE Alarm Type:=Multicast Communication Mismatch Alarm User Structure Identifier := 0x8000 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := Multicast Provider Status Mismatch Alarm.req ()	ID
31	ID	Alarm.cnf () => ignore	ID
32	ID	Read Output Data.ind () /ARUID==AREP.ARUID&& Slot==AREP.Slot Number&& Subslot==AREP.Subslot Number&& ((Length Output Data+Length IOPS+Length IOCS) <=Length) => AREP:=local AR reference Length:=Slot.Subslot.Length Data+ Slot.Subslot.Length IOCS+Slot.Subslot.Length IOPS Length Data:=Slot.Subslot.Length Data Length IOCS:=Slot.Subslot.Length IOCS Length IOPS:=Slot.Subslot.Length IOPS IOCS:=Slot.Subslot.IOCS IOPS:=Slot.Subslot.IOPS Subslot Output Data:=Output Data object subelement value Read Output Data.rsp (+)	ID

#	Current State	Event /Condition =>Action	Next State
33	ID	Read Output Data.ind () /(ARUUID==AREP.ARUUID&& Slot==AREP.Slot Number&& Subslot==AREP.Subslot Number&& ((Length Output Data+Length IOPS+Length IOCS) <=Length)) => AREP:=local AR reference Error Decode:=PNIORW Error Code 1:="parameter error" Error Code 2:=vendor specific or 0 Add Data 1:=profile specific or 0 Add Data 2:=vendor specific or 0 Read Output Data.rsp (-)	ID
34	ID	/input value changed => For all involved slot/subslots do: Slot := Slot Number; Subslot := Subslot Number IOPS := GOOD Subslot Input Data:=Input Data object subelement value For all involved slot/subslots do: Local Set Input_req ()	IDWC
35	ID	Read Input Data.ind () /ARUUID==AREP.ARUUID&& Slot==AREP.Slot Number&& Subslot==AREP.Subslot Number&& ((Length Input Data+Length IOPS+Length IOCS) <=Length) => AREP:=local AR reference Length:=Slot.Subslot.Length Data+ Slot.Subslot.Length IOCS+Slot.Subslot.Length IOPS Length Data:=Slot.Subslot.Length Data Length IOCS:=Slot.Subslot.Length IOCS Length IOPS:=Slot.Subslot.Length IOPS IOCS:=Slot.Subslot.IOCS IOPS:=Slot.Subslot.IOPS Subslot Input Data:=Input Data object subelement value Read Input Data.rsp (+)	ID
36	ID	Read Input Data.ind () /(ARUUID==AREP.ARUUID&& Slot==AREP.Slot Number&& Subslot==AREP.Subslot Number&& ((Length Input Data+Length IOPS+Length IOCS) <=Length)) => AREP:=local AR reference Error Decode:=PNIORW Error Code 1:="parameter error" Error Code 2:=vendor specific or 0 Add Data 1:=profile specific or 0 Add Data 2:=vendor specific or 0 Read Input Data.rsp (-)	ID

7.4.2.6 Functions, Macros, Timers and Variables

Table 284 contains the functions, macros, timers and variables used by DEVSM and their arguments and their descriptions.

Table 284 – Functions, Macros, Timers and Variables by DEVSM

Name	Type	Function/Meaning
CHECK_IOPS(CREP)	Function	all IO data objects of CREP dispose of IOPS==GOOD
UserApplReady	Function	This local function is issued from the user after all parameterization is done and input data is updated

SumARinData	Function	This local function is issued from DEVSM if all consumer reaches the state Data
DeltaConfiguration	Variable	Is the difference between the expected and real configuration
StoredSessionKey	Variable	This local variable stores the session key from the connect service
SetConnectResponseData	Function	This local function configures the module difference block for the connect response using the delta configuration
AdaptConfiguration	Function	This local function adapt the IO data layout of real configuration to the expected configuration
StoreParameter	Function	This local function stores the parameters with respect to the persistent flag
SetWrErrorResponseData	Function	This local function configures the response negative PDU
MCR_IOPS_BAD	Variable	This variable is used for the delta alarm generation of the multicast communication relation diagnosis

RecordDataObject are stored by the IO device. When receiving the Prm Endind a “full” RecordDataObjects check (locally defined rules) is done by the application. If the RecordDataObjects suit, they are stored by the application. This may already be done by Prm End.rsp or even later. From an “outside” view the takeover of the RecordDataObjects shall be done with ApplicationReady.req.

7.4.3 Physical Device parameter check

7.4.3.1 General

A check of the remote system data shall be done during startup and operation. The real remote system data of the peer connected to a port are retrieved by LLDP requests. So, the check of remote system data shall be done after every reception of a LLDP indication.

To run RT_CLASS_3 communication relationships between devices within a RT_CLASS_3 domain, the conformity between the engineered und the real peers of every device within the domain shall be guaranteed.

7.4.3.2 Remote system data

7.4.3.2.1 Common

Deviations of the remote system data are notified by alarm notifications transferred to the IO controller. For each port a state machine shall exist for neighborhood check (REM_CHK).

7.4.3.2.2 Primitive definitions

7.4.3.2.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by REM_CHK are described in the service definition and shown in Table 285.

Table 285 – Remote primitives issued or received by REM_CHK

Primitive	Source	Destination	Associated parameters	Functions
Alarm.cnf ()	FAL	REM_CHK	see 7.3.5.3	—
Alarm.req	REM_CHK	FAL	see 7.3.5.3	—

7.4.3.2.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by REM_CHK are described in the service definition and shown in Table 286.

Table 286 – Local primitives issued or received by REM_CHK

Primitive	Source	Destination	Associated parameters	Functions
Remote Systems Data Change.ind	local	REM_CHK	see 6.3.7.3.1	This local function is issued on every change of stored neighbors information

7.4.3.2.3 State transition diagram

Figure 46 shows the state diagram for the selected behavior.

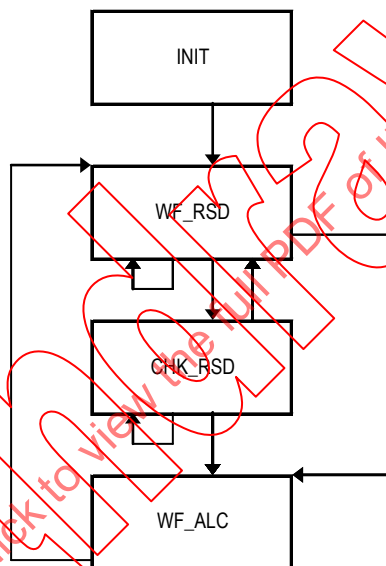


Figure 46 – State transition diagram REM_CHK

States of REM_CHK

INIT	Initial state
WF_RSD	Wait for remote systems data change
CHK_RSD	Check the remote systems data
WF_ALC	Wait for alarm confirmation

7.4.3.2.4 State machine description

This state machine waits for Remote Systems Data Change indications. If the indication contains information, which is in contradiction to the configured neighborhood information (such as neighbor, MAU type, cable delay...) or fits again the expected configuration an appropriate remote system data mismatch alarm (coming or going) is raised.

7.4.3.2.5 State table

Table 287 defines the state table for the neighborhood check.

Table 287 – State table REM_CHK

#	Current State	Event /Condition =>Action	Next State
1	INIT	=> REMOTE_DATA_OK := TRUE REMOTE_CABLE_DELAY_OK := TRUE REMOTE_PTCP_DATA_OK := TRUE REMOTE_MAUTYPE_OK := TRUE REMOTE_MRP_DOMAIN_OK := TRUE //NOTE To reduce the amount of diagnosis, "Remote mismatch::No peer detected" is used as an accumulative event in case of "Data transmission impossible::Port State Mismatch – Link down" or Remote System Data Change(Local Port ID, NO_DATA_AVAILABLE).	WF_RSD
2	WF_RSD	Remote Systems Data Change.ind () /engineered number of neighbors == NOT_ENGINEERED => ignore	WF_RSD
3	WF_RSD	Remote Systems Data Change.ind () /List of Neighbors == 0 && engineered number of neighbors == 0 && REMOTE_DATA_OK == TRUE (do similar for REMOTE_CABLE_DELAY_OK, REMOTE_PTCP_DATA_OK , REMOTE_MAUTYPE_OK, REMOTE_MRP_DOMAIN_OK) => ignore	WF_RSD
4	WF_RSD	Remote Systems Data Change.ind () /List of Neighbors == 0 && engineered number of neighbors == 0 && REMOTE_DATA_OK == FALSE do similar for REMOTE_CABLE_DELAY_OK, REMOTE_PTCP_DATA_OK , REMOTE_MAUTYPE_OK, REMOTE_MRP_DOMAIN_OK) => REMOTE_DATA_OK := TRUE CONVERT(Local Port ID) Alarm Type := Port Data Change Notification User Structure Identifier := 0x8000 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := REMOTE_SYSTEM_DATA_MISMATCH Alarm.req ()	WF_ALC
5	WF_RSD	Remote Systems Data Change.ind () /List of Neighbors > 0 && List of Neighbors == engineered number of neighbors => ignore	CHK_RSD

#	Current State	Event /Condition =>Action	Next State
6	WF_RSD	Remote Systems Data Change.ind () /engineered number of neighbors > 0 && List of Neighbors != engineered number of neighbors && REMOTE_DATA_OK == TRUE => REMOTE_DATA_OK := FALSE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Appears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := REMOTE_SYSTEM_DATA_MISMATCH Alarm.req ()	WF_ALC
7	WF_RSD	OTHERS => ignore	WF_RSD
8	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote Chassis ID, Remote Port ID) == OK && REMOTE_DATA_OK == FALSE => REMOTE_DATA_OK := TRUE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Disappears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := REMOTE_SYSTEM_DATA_MISMATCH Alarm.req ()	WF_ALC
9	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote Chassis ID, Remote Port ID) == OK && REMOTE_DATA_OK == TRUE => ignore	WF_RSD
10	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote Chassis ID, Remote Port ID) != OK && REMOTE_DATA_OK == FALSE => ignore	WF_RSD

#	Current State	Event /Condition =>Action	Next State
11	CHK_RSD	/CHK_REMOTE_DATA(List of Neighbors, Remote Chassis ID, Remote Port ID) != OK && REMOTE_DATA_OK == TRUE => REMOTE_DATA_OK := FALSE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Appears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := REMOTE_SYSTEM_DATA_MISMATCH Alarm.req ()	WF_ALC
12	CHK_RSD	/CHK_REMOTE_DATA(List of Neighbors, Remote Cable Delay) == OK && REMOTE_CABLE_DELAY_OK == FALSE => REMOTE_CABLE_DELAY_OK := TRUE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Disappears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_CABLE_DELAY_MISMATCH Alarm.req ()	WF_ALC
13	CHK_RSD	/CHK_REMOTE_DATA(List of Neighbors, Remote Cable Delay) == OK && REMOTE_CABLE_DELAY_OK == TRUE => ignore	WF_RSD
14	CHK_RSD	/CHK_REMOTE_DATA(List of Neighbors, Remote Cable Delay) != OK && REMOTE_CABLE_DELAY_OK == FALSE => ignore	WF_RSD
15	CHK_RSD	/CHK_REMOTE_DATA(List of Neighbors, Remote Cable Delay) != OK && REMOTE_CABLE_DELAY_OK == TRUE => REMOTE_CABLE_DELAY_OK := FALSE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Appears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_CABLE_DELAY_MISMATCH Alarm.req ()	WF_ALC

#	Current State	Event /Condition =>Action	Next State
16	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote PTCP Data) == OK && REMOTE_PTCP_DATA_OK == FALSE => REMOTE_PTCP_DATA_OK := TRUE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Disappears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_PTCP_MISMATCH Alarm.req ()	WF_ALC
17	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote PTCP Data) == OK && REMOTE_PTCP_DATA_OK == TRUE => ignore	WF_RSD
18	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote PTCP Data) != OK && REMOTE_PTCP_DATA_OK == FALSE => ignore	WF_RSD
19	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote PTCP Data) != OK && REMOTE_PTCP_DATA_OK == TRUE => REMOTE_PTCP_DATA_OK := FALSE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Appears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_PTCP_MISMATCH Alarm.req ()	WF_ALC
20	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MAU Data) == OK && REMOTE_MAUTYPE_OK == FALSE => REMOTE_MAUTYPE_OK := TRUE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Disappears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_MAUTYPE_MISMATCH Alarm.req ()	WF_ALC
21	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MAU Data) == OK && REMOTE_MAUTYPE_OK == TRUE => ignore	WF_RSD

#	Current State	Event /Condition =>Action	Next State
22	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MAU Data) != OK && REMOTE_MAUTYPE_OK == FALSE => ignore	WF_RSD
23	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MAU Data) != OK && REMOTE_MAUTYPE_OK == TRUE => REMOTE_MAUTYPE_OK := FALSE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Appears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_MAUTYPE_MISMATCH Alarm.req ()	WF_ALC
24	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MRP Data) == OK && REMOTE_MRP_DOMAIN_OK == FALSE => REMOTE_MRP_DOMAIN_OK := TRUE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Disappears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_MRP_DOMAIN_MISMATCH Alarm.req ()	WF_ALC
25	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MRP Data) == OK && REMOTE_MRP_DOMAIN_OK == TRUE => ignore	WF_RSD
26	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MRP Data) != OK && REMOTE_MRP_DOMAIN_OK == FALSE => ignore	WF_RSD
27	CHK_RSD	/CHECK_REMOTE_DATA(List of Neighbors, Remote MRP Data) != OK && REMOTE_MRP_DOMAIN_OK == TRUE => REMOTE_MRP_DOMAIN_OK := FALSE CONVERT(Local Port ID) Alarm Priority := ALARM_HIGH or ALARM_LOW Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties := "Appears" ChannelErrorType := REMOTE_MISMATCH ExtChannelErrorType := PEER_MRP_DOMAIN_MISMATCH Alarm.req ()	WF_ALC

#	Current State	Event /Condition =>Action	Next State
28	CHK_RSD	OTHERS => ignore	CHK_RSD
29	WF_ALC	Alarm.cnf () =>	WF_RSD

7.4.3.2.6 Functions, Macros, Timers and Variables

Table 288 contains the functions, macros, timers and variables used by REM_CHK and their arguments and their descriptions.

Table 288 – Functions, Macros, Timers and Variables by REM_CHK

Name	Type	Function/Meaning
CHECK_REMOTE_DATA	Function	This local function compares real Remote Data with engineered Remote Data
CONVERT	Function	This local function converts the Local Port ID to Slot Number and Subslot Number
REMOTE_SYSTEM_DATA_MISMATCH	Variable	Generates the corresponding ExtChannelErrorType for ChannelErrorType "Remote mismatch"
NO_DATA_AVAILABLE	Macro	List of neighbors is zero

7.4.3.3 Local system data

7.4.3.3.1 Common

Also the physical device parameters MAUType and Line Delay of every port shall be checked with the transition to the link up state of a specific port.

Deviations of the physical device parameters MAUType and Line Delay are notified by alarm notifications transferred to the IO controller. As a local result the forwarding of frames with frame ID range 3 (see Table 276) shall be stopped for this port. For each individual port a state machine shall exist for physical device parameter check.

7.4.3.3.2 Primitive definitions

7.4.3.3.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by local link state machine (LOC_LNK) are described in the service definition and shown in Table 289.

Table 289 – Remote primitives issued or received LOC_LNK

Primitive	Source	Destination	Associated parameters	Functions
Alarm.cnf ()	FAL	LOC_LNK	see 7.3.5.3	—
Alarm.req	LOC_LNK	FAL	see 7.3.5.3	—

7.4.3.3.2.2 Primitives exchanged between local machines

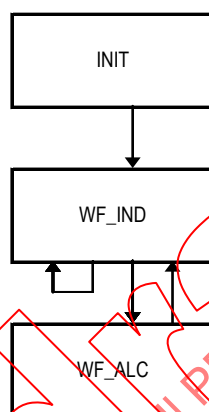
The local service primitives including their associated parameters issued or received by LOC_LNK are described in the service definition and shown in Table 290.

Table 290 – Local primitives issued or received LOC_LNK

Primitive	Source	Destination	Associated parameters	Functions
Line Delay Change.ind	local	LOC_LNK	see 6.3.2.3.8	—
MAU Type Change.ind	local	LOC_LNK	see 6.3.11.3.2	—
Port State Change.ind	local	LOC_LNK	see 6.3.9.3.1	—

7.4.3.3.3 State transition diagram

Figure 47 shows the state diagram for the selected behavior.

**Figure 47 – State transition diagram LOC_LNK**

States of the LOC_LNK

INIT	Initial state
WF_IND	Wait for a change indication and issue an alarm if necessary
WF_ALC	Wait for alarm confirmation

7.4.3.3.4 State machine description

This state machine works on port granularity and awaits MAU type change, port state change and line delay change indications of the local port. If the change indication is in contradictory to the local port configuration or fulfills it again, an appropriate Data Transmission Impossible alarm (coming or going) is raised.

7.4.3.3.5 State table

Table 291 defines the state table for the selected behavior.

Table 291 – State table LOC_LNK

#	Current State	Event /Condition =>Action	Next State
1	INIT	=> MAUTYPE_LINK_OK := FALSE PORT_STATE_OK := FALSE LINE_DELAY_OK := FALSE (PORT_STATE_OK has a strong dependency to the usage of MRP. If MRP is activated, the steering of the port state is part of the MRP protocol. In this case the state BLOCKED of a MRM may be similar handled to the state FORWARDING if it is signaled from a ring port.)	WF_IND
2	WF_IND	MAU Type Change.ind () /Local Link Status == Link_Ok && CHECK_MAUTYPE(Local MAU Type) == OK && PORT_STATE_OK == TRUE && LINE_DELAY_OK == TRUE => CONVERT(Local Port ID) MAUTYPE_LINK_OK := TRUE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8000 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE Alarm.req ()	WF_ALC
3	WF_IND	MAU Type Change.ind () /Local Link Status == Link_Ok && CHECK_MAUTYPE(Local MAU Type) == OK && (PORT_STATE_OK == FALSE LINE_DELAY_OK == FALSE) => MAUTYPE_LINK_OK := TRUE	WF_IND
4	WF_IND	MAU Type Change.ind () /Local Link Status == Link_Ok && (CHECK_MAUTYPE(Local MAU Type) != OK PORT_STATE_OK == FALSE LINE_DELAY_OK == FALSE) => CONVERT(Local Port ID) Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := LINK_DOWN Alarm.req ()	WF_ALC

#	Current State	Event /Condition =>Action	Next State
5	WF_IND	MAU Type Change.ind () /CHECK_MAUTYPE(Local MAU Type) == OK && (Local Link Status != Link_Ok PORT_STATE_OK == FALSE LINE_DELAY_OK == FALSE) => CONVERT(Local Port ID) Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := MAUTYPE_MISMATCH Alarm.req ()	WF_ALC
6	WF_IND	MAU Type Change.ind () /Local Link Status == Link_Fail => CONVERT(Local Port ID) MAUTYPE_LINK_OK := FALSE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Appears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := LINK_DOWN Alarm.req ()	WF_ALC
7	WF_IND	MAU Type Change.ind () /CHECK_MAUTYPE(Local MAU Type) != OK => CONVERT(Local Port ID) MAUTYPE_LINK_OK := FALSE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Appears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := MAUTYPE_MISMATCH Alarm.req ()	WF_ALC
8	WF_IND	MAU Type Change.ind () /Local Link Status == Link_Ready => MAUTYPE_LINK_OK := FALSE	WF_IND
9	WF_IND	Port State Change.ind () /Local Port Status == Port_Forwarding && LINE_DELAY_OK == TRUE && MAUTYPE_LINK_OK == TRUE => CONVERT(Local Port ID) PORT_STATE_OK := TRUE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8000 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE Alarm.req ()	WF_ALC

#	Current State	Event /Condition =>Action	Next State
10	WF_IND	Port State Change.ind () /Local Port Status == Port_Disable ==> CONVERT(Local Port ID) PORT_STATE_OK := FALSE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Appears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := PORTSTATE_MISMATCH Alarm.req ()	WF_ALC
11	WF_IND	Port State Change.ind () /Local Port Status == Port_Forwarding && (PORT_STATE_OK == FALSE MAUTYPE_LINK_OK == FALSE) ==> CONVERT(Local Port ID) PORT_STATE_OK := TRUE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := PORTSTATE_MISMATCH Alarm.req ()	WF_ALC
12	WF_IND	Port State Change.ind () /Local Port Status == Port_Discarding ==> PORT_STATE_OK := FALSE	WF_IND
13	WF_IND	Port State Change.ind () /Local Port Status == Port_Learning ==> PORT_STATE_OK := FALSE	WF_IND
14	WF_IND	Line Delay Change.ind () /CHECK_LINE_DELAY(Line Delay) == Line Delay is equal or less than engineered Line Delay && MAUTYPE_LINK_OK == TRUE && PORT_STATE_OK == TRUE ==> CONVERT(Local Port ID) LINE_DELAY_OK == TRUE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8000 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE Alarm.req ()	WF_ALC

#	Current State	Event /Condition =>Action	Next State
15	WF_IND	Line Delay Change.ind () /CHECK_LINE_DELAY(Line Delay) == Line Delay is larger than engineered Line Delay ==> CONVERT(Local Port ID) LINE_DELAY_OK == FALSE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Appears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := LINE_DELAY_MISMATCH Alarm.req ()	WF_ALC
16	WF_IND	Line Delay Change.ind () /CHECK_LINE_DELAY(Line Delay) == Line Delay is equal or less than engineered Line Delay && (MAUTYPE_LINK_OK == FALSE PORT_STATE_OK == FALSE) ==> CONVERT(Local Port ID) LINE_DELAY_OK == TRUE Alarm Type := Port Data Change Notification User Structure Identifier := 0x8002 ChannelNumber := 0x8000 ChannelProperties.Specifier := Disappears ChannelErrorType := DATA_TRANSMISSION_IMPOSSIBLE ExtChannelErrorType := LINE_DELAY_MISMATCH Alarm.req ()	WF_ALC
17	WF_IND	OTHERS ignore ==> ignore	WF_IND
18	WF_ALC	Alarm.cnf () ==>	WF_IND

7.4.3.3.6 Functions, Macros, Timers and Variables

Table 292 contains the functions, macros, timers and variables used by the LOC_LNK and their arguments and their descriptions.

Table 292 – Functions, Macros, Timers and Variables used by LOC_LNK

Name	Type	Function/Meaning
CHECK_LINE_DELAY	Function	Compare Line Delay with engineered line delay
CHECK_MAUTYPE	Function	Compare Local MAU type with engineered MAU type
CONVERT	Function	Convert Local Port ID to Slot Number and Subslot Number
EXTENDCHANNELDIAGNOSIS	Macro	0x8002 - data is transferred in extended diagnosis channel format
SUBMODULE	Macro	0x8000 - refers to the submodule
LINE_DELAY_OK	VARIABLE	Local variable holding the result of line delay comparison
MAUTYPE_LINK_OK	VARIABLE	Local variable holding the result of MAU type comparison
PORT_STATE_OK	VARIABLE	Local variable holding the port state

7.4.3.4 Optical system data

7.4.3.4.1 Common

The physical device parameters Power Budget of every optical port shall be checked in a cycle of 1 s.

Deviations of the physical device parameters Power Budget are notified by alarm notifications transferred to the IO controller.

However, only if the used hardware for fiber optic supports reading the power budget, the support of diagnosis and maintenance information is mandatory.

7.4.3.4.2 Fiber optic maintenance required

7.4.3.4.2.1 Common

For each individual port a state machine for fiber optic maintenance required (FOMR) shall exist.

7.4.3.4.2.2 Primitive definitions

7.4.3.4.2.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by FOMR are described in the service definition and shown in Table 293.

Table 293 – Remote primitives issued or received by FOMR

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	FOMR	FAL	see 7.3.5.3	—

7.4.3.4.2.2.2 Primitives exchanged between local machines

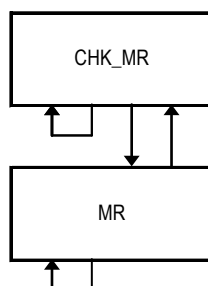
The local service primitives including their associated parameters issued or received by FOMR are described in the service definition and shown in Table 294.

Table 294 – Local primitives issued or received by FOMR

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.4.3.4.2.3 State transition diagram

The state transition diagram is shown in Figure 48.

**Figure 48 – State transition diagram FOMR****States of the FOMR****CHK_MR**

Check the maintenance required value against the actual value, issue an alarm if lower and change to MR

MR

Check the maintenance required value against the actual value, issue an alarm if higher and change to MR

7.4.3.4.2.4 State machine description

This state machine on port granularity checks every second whether the actual budget of the optical port drops beneath the configured maintenance required threshold. If being so, a coming maintenance required Fiber Optic Mismatch / Power Budget alarm is raised. If the power budget raises again over the threshold, a going alarm is raised.

7.4.3.4.2.5 State table

The state table is shown in Table 295.

Table 295 – State table FOMR

#	Current State	Event /Condition =>Action	Next State
1	CHK_MR	Init => StartTimer (1 s)	CHK_MR
2	CHK_MR	TimerExpired Maintenance Required Power Budget = DISABLED Current Power Budget > Maintenance Required Power Budget => StartTimer (1 s)	CHK_MR

#	Current State	Event /Condition =>Action	Next State
3	CHK_MR	TimerExpired /Maintenance Required Power Budget != DISABLED && Current Power Budget <= Maintenance Required Power Budget => StartTimer (1 s) Alarm Priority:=ALARM_HIGH or ALARM_LOW Alarm type:=Diagnosis User Structure Identifier := 0x8100 ChannelNumber := 0x8000 ChannelProperties.Maintenance Required := MAINTENANCE_REQUIRED ChannelProperties.Maintenance Demanded := NO_MAINTENANCE_DEMANDED Channel Properties.Specifier := APPEARS ChannelProperties.type := UNSPECIFIC ChannelProperties.Accumulative := INDIVIDUAL ChannelProperties.Direction := MANUFACTURER_SPECIFIC ChannelErrorType := FIBER_OPTIC_MISMATCH ExtChannelErrorType := POWER_BUDGET ExtChannelAddValue := Current Power Budget Alarm.req ()	MR
4	MR	TimerExpired / Current Power Budget <= Maintenance Required Power Budget => StartTimer (1 s)	MR
5	MR	TimerExpired /Current Power Budget > Maintenance Required Power Budget Maintenance Required Power Budget != DISABLED => StartTimer (1 s) Alarm Priority:=ALARM_HIGH or ALARM_LOW Alarm type:=Diagnosis User Structure Identifier := 0x8100 ChannelNumber := 0x8000 ChannelProperties.Maintenance Required := MAINTENANCE_REQUIRED ChannelProperties.Maintenance Demanded := NO_MAINTENANCE_DEMANDED Channel Properties.Specifier := DISAPPEARS ChannelProperties.type := UNSPECIFIC ChannelProperties.Accumulative := INDIVIDUAL ChannelProperties.Direction := MANUFACTURER_SPECIFIC ChannelErrorType := FIBER_OPTIC_MISMATCH ExtChannelErrorType := POWER_BUDGET ExtChannelAddValue := Current Power Budget Alarm.req ()	CHK_MR

7.4.3.4.2.6 Functions, Macros, Timers and Variables

Table 296 contains the functions, macros, timers and variables used by FOMR and their arguments and their descriptions.

Table 296 – Functions, Macros, Timers and Variables used by FOMR

Name	Type	Function/Meaning
StartTimer	Function	This local function starts or restarts a timer
TimerExpired	Function	This local function is issued if a timer expires
Maintenance Required Power Budget	Variable	This local variable contains the threshold value for maintenance required
Current Power Budget	Variable	This local variable contains the actual power value.

7.4.3.4.3 Fiber optic maintenance demanded

7.4.3.4.3.1 Common

For each individual port a state machine for fiber optic maintenance demanded (FOMD) shall exist.

7.4.3.4.3.2 Primitive definitions

7.4.3.4.3.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by FOMD are described in the service definition and shown in Table 297.

Table 297 – Remote primitives issued or received by FOMD

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	FOMD	FAL	see 7.3.5.3	—

7.4.3.4.3.2.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by FOMD are described in the service definition and shown in Table 298.

Table 298 – Local primitives issued or received by FOMD

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.4.3.4.3.3 State transition diagram

The state transition diagram is shown in Figure 49.

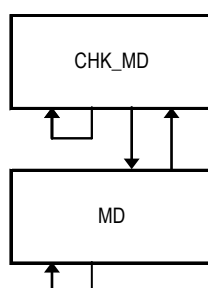


Figure 49 – State transition diagram FOMD

States of the FOMD

CHK_MD

Check the maintenance demanded value against the actual value, issue an alarm if lower and change to MD

MD

Check the maintenance demanded value against the actual value, issue an alarm if higher and change to MD

7.4.3.4.3.4 State machine description

This state machine on port granularity checks every second whether the actual budget of the optical port drops beneath the configured maintenance demanded threshold. If being so, a coming maintenance demanded Fiber Optic Mismatch / Power Budget alarm is raised. If the power budget raises again over the threshold, a going alarm is raised.

7.4.3.4.3.5 State table

The state table is shown in Table 299.

Table 299 – State table FOMD

#	Current State	Event /Condition =>Action	Next State
1	CHK_MD	Init => StartTimer (1 s)	CHK_MD
2	CHK_MD	TimerExpired /Maintenance Demanded Power Budget = DISABLED Current Power Budget > Maintenance Demanded Power Budget => StartTimer (1 s)	CHK_MD
3	CHK_MD	TimerExpired /Maintenance Demanded Power Budget != DISABLED && Current Power Budget <= Maintenance Demanded Power Budget => StartTimer (1 s) Alarm Priority:=ALARM_HIGH or ALARM_LOW Alarm type:=Diagnosis User Structure Identifier := 0x8100 ChannelNumber := 0x8000 ChannelProperties.Maintenance Required := NO_MAINTENANCE_REQUIRED ChannelProperties.Maintenance Demanded := MAINTENANCE_DEMANDED ChannelProperties.Specifier := APPEARS ChannelProperties.type := UNSPECIFIC ChannelProperties.Accumulative := INDIVIDUAL ChannelProperties.Direction := MANUFACTURER_SPECIFIC ChannelErrorType := FIBER_OPTIC_MISMATCH ExtChannelErrorType := POWER_BUDGET ExtChannelAddValue := Current Power Budget Alarm.req ()	MD
4	MD	TimerExpired /Current Power Budget <= Maintenance Demanded Power Budget => StartTimer (1 s)	MD
5	MD	TimerExpired /Current Power Budget > Maintenance Demanded Power Budget Maintenance Demanded Power Budget != DISABLED => StartTimer (1 s) Alarm Priority:=ALARM_HIGH or ALARM_LOW Alarm type:=Diagnosis User Structure Identifier := 0x8100 ChannelNumber := 0x8000 ChannelProperties.Maintenance Required := NO_MAINTENANCE_REQUIRED ChannelProperties.Maintenance Demanded := MAINTENANCE_DEMANDED ChannelProperties.Specifier := DISAPPEARS ChannelProperties.type := UNSPECIFIC ChannelProperties.Accumulative := INDIVIDUAL ChannelProperties.Direction := MANUFACTURER_SPECIFIC ChannelErrorType := FIBER_OPTIC_MISMATCH ExtChannelErrorType := POWER_BUDGET ExtChannelAddValue := Current Power Budget Alarm.req ()	CHK_MD

7.4.3.4.3.6 Functions, Macros, Timers and Variables

Table 300 contains the functions, macros, timers and variables used by FOMD and their arguments and their descriptions.

Table 300 – Functions, Macros, Timers and Variables by FOMD

Name	Type	Function/Meaning
StartTimer	Function	This local function starts or restarts a timer
TimerExpired	Function	This local function is issued if a timer expires
Maintenance Demanded Power Budget	Variable	This local variable contains the threshold value for maintenance demanded
Current Power Budget	Variable	This local variable contains the actual power value.

7.4.3.4.4 Fiber optic diagnosis

7.4.3.4.4.1 Common

For each individual port a state machine for fiber optic diagnosis (FODIAG) shall exist.

7.4.3.4.4.2 Primitive definitions

7.4.3.4.4.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by FODIAG are described in the service definition and shown in Table 301.

Table 301 – Remote primitives issued or received by FODIAG

Primitive	Source	Destination	Associated parameters	Functions
Alarm.req	FODIAG	FAL	see 7.3.5.3	—

7.4.3.4.4.2.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by FODIAG are described in the service definition and shown in Table 302.

Table 302 – Local primitives issued or received by FODIAG

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.4.3.4.4.3 State transition diagram

The state transition diagram is shown in Figure 50.

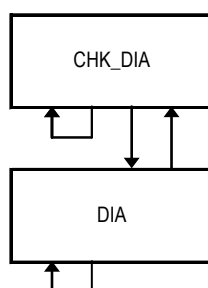


Figure 50 – State transition diagram FODIAG

States of the FODIAG

CHK_DIA

Check the error value against the actual value, issue an alarm if lower and change to DIA

DIA

Check the error value against the actual value, issue an alarm if higher and change to DIA

7.4.3.4.4 State machine description

This state machine on port granularity checks every second whether the actual budget of the optical port drops beneath the configured diagnosis threshold. If being so, a coming diagnosis Fiber Optic Mismatch / Power Budget alarm is raised. If the power budget raises again over the threshold, a going alarm is raised.

7.4.3.4.4.5 State table

The state table is shown in Table 303.

Table 303 – State table FODIAG

#	Current State	Event /Condition =>Action	Next State
1	CHK_DIA	Init => StartTimer (1 s)	CHK_DIA
2	CHK_DIA	TimerExpired / Error Power Budget = DISABLED Current Power Budget > Error Power Budget => StartTimer (1 s)	CHK_DIA

#	Current State	Event /Condition =>Action	Next State
3	CHK_DIA	TimerExpired / Error Power Budget != DISABLED && Current Power Budget <= Error Power Budget => StartTimer (1 s) Alarm Priority:=ALARM_HIGH or ALARM_LOW Alarm type:=Diagnosis User Structure Identifier := 0x8100 ChannelNumber := 0x8000 ChannelProperties.Maintenance Required := NO_MAINTENANCE_REQUIRED ChannelProperties.Maintenance Demanded := NO_MAINTENANCE_DEMANDED Channel Properties.Specifier := APPEARS ChannelProperties.type := UNSPECIFIC ChannelProperties.Accumulative := INDIVIDUAL ChannelProperties.Direction := MANUFACTURER_SPECIFIC ChannelErrorType := FIBER_OPTIC_MISMATCH ExtChannelErrorType := POWER_BUDGET ExtChannelAddValue := Current Power Budget Alarm.req ()	DIA
4	DIA	TimerExpired / Current Power Budget <= Error Power Budget => StartTimer (1 s)	DIA
5	DIA	TimerExpired / Current Power Budget > Error Power Budget Error Power Budget != DISABLED => StartTimer (1 s) Alarm Priority:=ALARM_HIGH or ALARM_LOW Alarm type:=Diagnosis User Structure Identifier := 0x8100 ChannelNumber := 0x8000 ChannelProperties.Maintenance Required := NO_MAINTENANCE_REQUIRED ChannelProperties.Maintenance Demanded := NO_MAINTENANCE_DEMANDED Channel Properties.Specifier := DISAPPEARS ChannelProperties.type := UNSPECIFIC ChannelProperties.Accumulative := INDIVIDUAL ChannelProperties.Direction := MANUFACTURER_SPECIFIC ChannelErrorType := FIBER_OPTIC_MISMATCH ExtChannelErrorType := POWER_BUDGET ExtChannelAddValue := Current Power Budget Alarm.req ()	CHK_DIA

7.4.3.4.4.6 Functions, Macros, Timers and Variables

Table 304 contains the functions, macros, timers and variables used by FODIAG and their arguments and their descriptions.

Table 304 – Functions, Macros, Timers and Variables by FODIAG

Name	Type	Function/Meaning
StartTimer	Function	This local function starts or restarts a timer
TimerExpired	Function	This local function is issued if a timer expires
Error Power Budget	Variable	This local variable contains the threshold value for error
Current Power Budget	Variable	This local variable contains the actual power value

7.4.4 Diagnosis and problem indicator

This subclause describes the application interaction within an IO device concerning diagnosis information.

It includes extracts of the behavior of the following ASEs:

- Diagnosis ASE
- IO Data ASE

There is a means to notify the presence of diagnosis information within IO data transfer referred to as “Problem Indicator” as part of the APDU status. The problem indicator shall be set as long as the Diagnosis ASE contain at least one diagnosis entry that affects the related AR. The IO data ASE provides services to set and get this parameter.

7.4.5 User alarms

This subclause describes the application interaction within an IO device concerning alarms that are caused by diagnosis, process, status, update or other manufacturer specific events. The notification of such alarms is manufacturer specific.

It includes extracts of the behavior of the following ASEs:

- Alarm ASE
- Diagnosis ASE (partly)

The Diagnosis Alarm is a means to notify the event if diagnosis information appears or disappears. It is strongly recommended to use channel diagnosis.

The general behavior of alarms is described in 7.3.4.4.

7.4.6 Behavior in case of configuration change

7.4.6.1 General

The behavior in case of pull or plug of modules and/or submodules, IO supervisor control and release during operation and AR establishment and release is referred to as configuration change.

The behavior in case of configuration change is defined as described below.

If a module is pulled the IOPS and IOCS for each subslot that belongs to the slot shall be set to BAD_BY_DEVICE. Furthermore, the value 0 for the parameter subslot implicitly covers all other subslots that belong to the slot in case of pull. If a submodule is pulled then a dedicated alarm notification shall be issued for the specific submodule. In any case the IO controller shall acknowledge the alarm.

7.4.6.2 State behavior of submodules

This subclause describes the application behavior of an IO device in case of plugging or pulling a module or submodule, change of parameterization data, if the IOPS or IOCS changes from BAD to GOOD, or if a submodule is released by an IO supervisor.

If there is more than one IO controller connected to a certain submodule and this module is e.g. released by an IO supervisor then always the first connected IO controller shall get the control back (first serve).

If a module is pulled then the pull module alarm shall be issued with the value zero for the parameter API.

It includes extracts of the behavior of the following ASEs:

- Alarm ASE
- Context ASE
- Record Data ASE
- IO Data ASE

7.4.6.2.1 Real Submodule

7.4.6.2.1.1 Primitive definitions

7.4.6.2.1.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by Real Submodule State Machine (RSMSM) are described in the service definition and shown in Table 305.

Table 305 – Remote primitives issued or received by RSMSM

Primitive	Source	Destination	Associated parameters	Functions
UserPlugSM.req	local	RSMSM	RealModuleID, RealSubmoduleID, API, Slot, Subslot	—
UserPullISM.req	local	RSMSM	API, Slot, Subslot	—
UserPlugSM.cnf ()	RSMSM	local		—
UserPullISM.cnf ()	RSMSM	local		—
PlugAlarm.req	RSMSM	PLUGSM	RealModuleID, RealSubmoduleID, API, Slot, Subslot	—
PlugAlarm.cnf ()	PLUGSM	RSMSM	API, Slot, Subslot	—
PlugWrongAlarm.req	RSMSM	PLUGSM	RealModuleID, RealSubmoduleID, API, Slot, Subslot	—
PlugWrongAlarm.cnf ()	PLUGSM	RSMSM	API, Slot, Subslot	—
PullAlarm.req	RSMSM	PULLSM	API, Slot, Subslot	—
PullAlarm.cnf ()	PULLSM	RSMSM	API, Slot, Subslot	—

7.4.6.2.1.1.2 Primitives exchanged between local machines

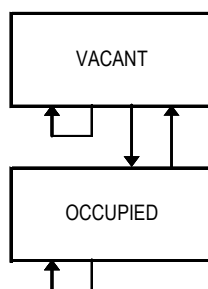
The local service primitives including their associated parameters issued or received by RSMSM described in the service definition and shown in Table 306.

Table 306 – Local primitives issued or received by RSMSM

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.4.6.2.1.2 State transition diagram

The state transition diagram is shown in Figure 51.

**Figure 51 – State transition diagram RSMSM****States of the RSMSM****VACANT**

The addressed API, Slot, Subslot is vacant. This means no submodule is plugged or assigned.

OCCUPIED

The addressed API, Slot, Subslot is occupied. This means a submodule is plugged or assigned.

7.4.6.2.1.3 State machine description

This state machine covers for a particular subslot its occupation by a submodule and raises the appropriate Plug and Unplug alarms. Note that the Plug alarm is not issued, when a wrong submodule is plugged into the subslot, if no current owner exists, or if the current owner is the application (SOL).

7.4.6.2.1.4 State table

Table 307 defines the state table for the selected behavior.

Table 307 – State table RSMSM

#	Current State	Event /Condition =>Action	Next State
1	VACANT	UserPlugSM.req () /GetOwnerFromAdministration () == FREE GetOwnerFromAdministration () == SOL ==> ignore UserPlugSM.cnf (+)	OCCUPIED
2	VACANT	UserPlugSM.req () /GetOwnerFromAdministration () == OWNER && UserCheckIdent () != WRONG ==> Create PlugSubmodule alarm PDU for the OWNER PlugAlarm.req () UserPlugSM.cnf (+)	OCCUPIED

#	Current State	Event /Condition =>Action	Next State
3	VACANT	UserPlugSM.req () /GetOwnerFromAdministration () == OWNER && UserCheckIdent () == WRONG => Create PlugWrongSubmodule alarm PDU for the OWNER PlugWrongAlarm.req () UserPlugSM.cnf (+)	OCCUPIED
4	VACANT	UserPullSM.req () => ignore UserPullSM.cnf (+)	VACANT
5	VACANT	PlugAlarm.cnf () => ignore	VACANT
6	VACANT	PlugWrongAlarm.cnf () => ignore	VACANT
7	VACANT	PullAlarm.cnf () => ignore	VACANT
8	OCCUPIED	UserPullSM.req () /GetOwnerFromAdministration () == FREE GetOwnerFromAdministration () == SOL => ignore UserPullSM.cnf (+)	VACANT
9	OCCUPIED	UserPullSM.req () /GetOwnerFromAdministration () == OWNER => Create PullSubmodule alarm PDU for the OWNER PullAlarm.req () UserPullSM.cnf (+)	VACANT
10	OCCUPIED	UserPlugSM.req () => ignore UserPlugSM.cnf (-)	OCCUPIED
11	OCCUPIED	PlugAlarm.cnf () => ignore	OCCUPIED
12	OCCUPIED	PlugWrongAlarm.cnf () => ignore	OCCUPIED
13	OCCUPIED	PullAlarm.cnf () => ignore	OCCUPIED

7.4.6.2.1.5 Functions, Macros, Timers and Variables

Table 308 contains the functions, macros, timers and variables used by RSMSM and their arguments and their descriptions.

Table 308 – Functions, Macros, Timers and Variables used by RSMSM

Name	Type	Function/Meaning

Name	Type	Function/Meaning
UserCheckIdent	Function	UserCheckIdent(API, Slot, Subslot, RealModuleID, RealSubmoduleID, ExpModuleID, ExpSubmoduleID) Returns the ident information for the expected submodule: OK: is returned, if a submodule is plugged at the specified location (API, slot and subslot) and it's the right one; i.e. its real ident information (RealModuleID, RealSubmoduleID) is equivalent to the expected ident information (ExpModuleID and ExpSubmoduleID). SUBSTITUTE: is returned, if a submodule is plugged at the specified location (API, slot and subslot) and is able to be a decent substitute; i.e. its real ident information (RealModuleID, RealSubmoduleID) is able to substitute the expected ident information (ExpModuleID and ExpSubmoduleID). WRONG: is returned, if a submodule is plugged at the specified location (API, slot and subslot) and it's wrong; i.e. if its real ident information (RealModuleID, RealSubmoduleID) does not match the expected ident information (ExpModuleID and ExpSubmoduleID), or if it doesn't support the expected API. NOSUBMODULE: is returned, if no submodule is plugged at the location specified by API, slot and subslot. using the following parameters: API, Slot, Subslot, RealModuleID, RealSubmoduleID, ExpModuleID, ExpSubmoduleID
GetOwnerFromAdministration	Function	This local function returns the owner of the submodule. FREE: No owner OWNER: The asking AREP is the Owner SOL: The submodule is superordinated locked

7.4.6.2.1.6 Truth Table

Table 309 contains the truth table used by RSMSM.

Table 309 – Rules for Submodule State.Ident Info

Input	Output - Owner	Output - Not owner
Result of UserCheckIdent()	Submodule State.Ident Info	Submodule State.Ident Info
OK	OK	OK
SUBSTITUTE	SUBSTITUTE	WRONG
WRONG	WRONG	WRONG
NOSUBMODULE	NOSUBMODULE	NOSUBMODULE

7.4.6.2.2 Ownership and Association of a Submodule

7.4.6.2.2.1 Common

For the access to a submodule within a particular AR there exist the following three "qualities":

- If the AR is the owner of the submodule, the AR has

- read access to input data,
 - write access to output data,
 - read access to data records,
 - write access to data records,
 - and is allowed to create or receive every kind of alarm notification.
- If the AR is associate of the submodule, the AR has
 - read access to input data,
 - read access to data records,
 - and is allowed to create or receive only the Return Of Submodule alarm notification.
 - A device access AR has read access to data records and if the application allows it also write access.
 - An implicit AR has read access to data records only.

The handling of the ownership for a particular submodule is described in Figure 52:

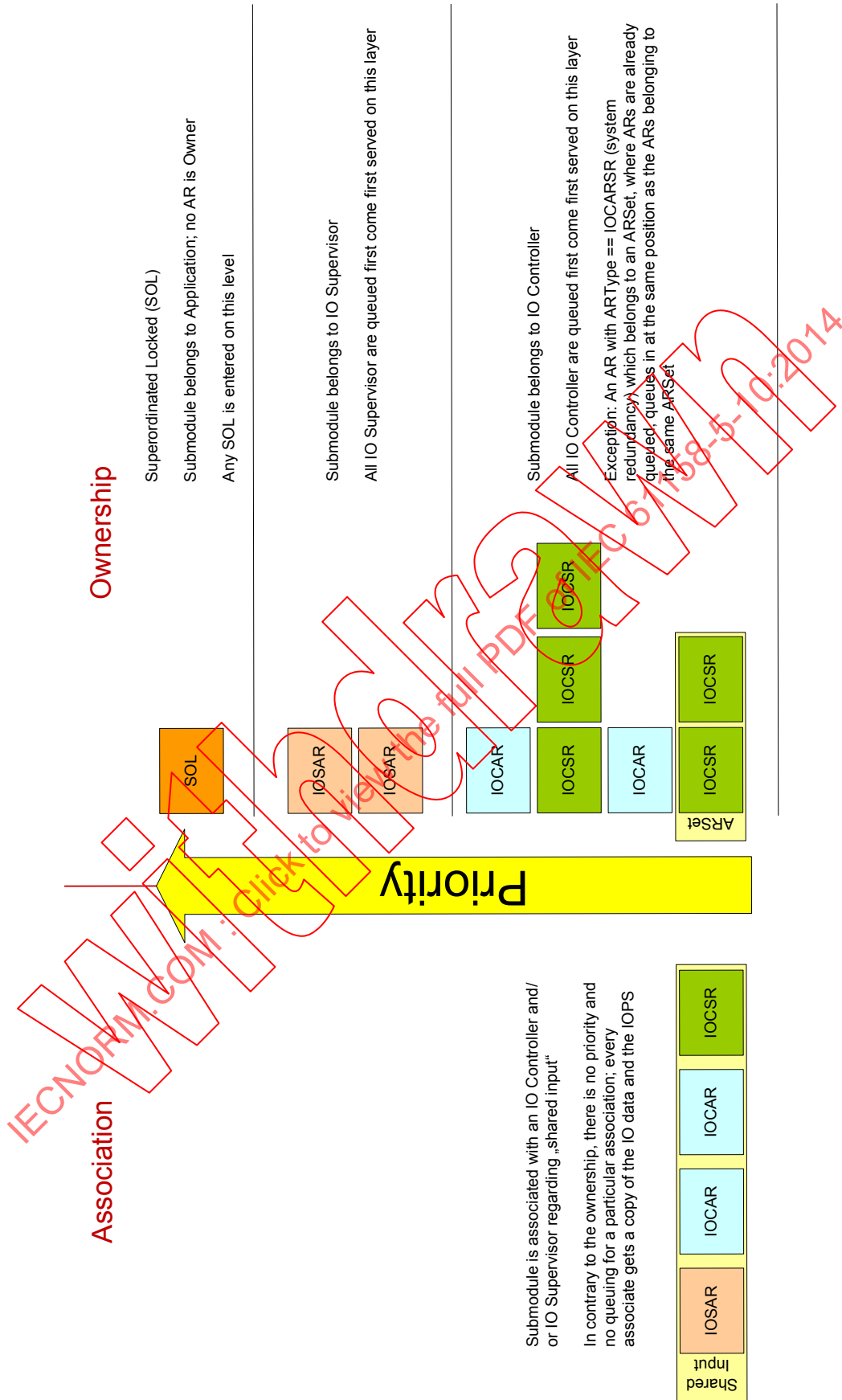


Figure 52 – Ownership handling

The ownership is based on the following principles:

- The ownership of a submodule is in principle handled “first come first served”. Thus the first application relation wanting the submodule will get the ownership; any further application relation will get a “Locked” and is queued at the end.
- The user application is able to lock the submodule for superordinated use at any time. The current owner of the submodule is informed by a “Controlled” alarm.
- If the current owner’s application relation allows a supervisor takeover (as indicated by the ARProperties), an application relation with ARTYPE IOSAR is able to takeover the submodule from its current owner. This application relation is set as the queue head and thus gets the ownership; the now previous owner is then informed by a “Controlled” alarm.
- If the current owner’s application relation doesn’t allow a supervisor takeover (as indicated by the ARProperties), an application relation with ARTYPE IOSAR is not able to takeover the submodule from its current owner. Thus the IO supervisor application relation is queued after the head but after the last waiting IO supervisor AR.
- A device access AR has no ownership on submodules; it is thus treated specially. It offers the ability to read or write data records to submodules. A read access is treated like an implicit read. For a write access the application decides whether the write access is allowed (i.e. not the ownership state machine).
- An application relation with ARTYPE IOCARSR of an already known AR Set (as indicated by the application relations ARUUID; details see /IEC 61158-6-10/) is queued in at the same location as the other ARs of this AR Set. Note that all application relations of the AR Set have the same view on the ownership as expressed by the ModuleDiffBlock.
- If the current owner goes, the ownership is handed over to the next application relation waiting in the queue. If the new owner is an application relation with ARTYPE != IOCARSR, a Released alarm is issued to indicate the hand over. If the ARTYPE is IOCARSR and the new owner is in the same AR Set as the previous owner, no Released alarm is issued.
- The ownership is handled in the OWNISM.

The association is based on the following principles:

- The association of a submodule is handled without any priority. All requesting ARs have the same priority; there is no queuing.
- The reason to associate a submodule is to have read access using an I-CR to the input data of a submodule from another IO controller / IO supervisor. This is also called “shared input”.
- Since the validity of the input data is determined by the IOPS, the Return of Submodule is a necessary alarm for an IO controller using an association.
- Since no alarms are allowed on an association beside the Return Of Submodule alarm, an associate has only information about the availability of the input data. Other information like the submodule’s presence (pull, plug, plug wrong), its parameterization state (application ready pending) or its diagnosis of the submodule is not available, since it will not be actualized via some alarm. Thus the Submodule State shall contain in any Module Diff Block for a shared input submodule
 - for the Ident Info (indicating its presence) the value OK,
 - for Qualified Info (indicating its qualified information) the value NO_QUALIFIED_INFO,
 - for Maintenance Required (indicating its maintenance required information) the value NO_MAINTENANCE_REQUIRED,
 - for Maintenance Demanded (indicating its maintenance demanded information) the value NO_MAINTENANCE_DEMANDED,
 - for Diag Info (indicating its diagnosis information) the value NO_DIAG,
 - for ARInfo (indicating ownership problems) the value OWN,

- independent of the submodule's state. In other words this information is only available for the application having the ownership on the submodule. The rules for setting the IOPS to BAD e.g. in case of a pulled or a wrongly plugged submodule (see 7.3.2.2) will take care that the submodule's input data is not useable in such a case.

– The association is handled in the ASSSM.

7.4.6.2.2.2 Primitive definitions

7.4.6.2.2.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by Ownership Submodule State Machine (OWNSM) and Association Submodule State Machine (ASSSM) are described in the service definition and shown in Table 310.

Table 310 – Remote primitives issued or received by OWNSM and ASSSM

Primitive	Source	Destination	Associated parameters	Functions
Connect.req	FAL	OWNSM, ASSSM	see 7.3.6.3.1	—
Read.req	FAL	OWNSM, ASSSM	see 7.3.1.4.1.1	—
Release.req	FAL	OWNSM, ASSSM	see 7.3.6.3.2	—
Write.req	FAL	OWNSM, ASSSM	see 7.3.1.4.2.1	—
UserSetSMState.cnf	local	OWNSM	API, Slot, Subslot	—
UserSetSMState.req	local	OWNSM	Command, API, Slot, Subslot	—
Read.rsp	OWNSM, ASSSM	FAL	see 7.3.1.4.1.1	—
Write.rsp	OWNSM, ASSSM	FAL	see 7.3.1.4.2.1	—
ReleasedAlarm.req	OWNSM	PLUGSM	ANY.AREP, ANY.API, ANY.Slot, ANY.Subslot	—
ControlledAlarm.req	OWNSM	PULLSM	ANY.AREP, ANY.API, ANY.Slot, ANY.Subslot	—
ReleasedAlarm.cnf	PLUGSM	OWNSM	ANY.AREP, ANY.API, ANY.Slot, ANY.Subslot	—
ControlledAlarm.cnf	PULLSM	OWNSM	ANY.AREP, ANY.API, ANY.Slot, ANY.Subslot	—

7.4.6.2.2.2.2 Primitives exchanged between local machines

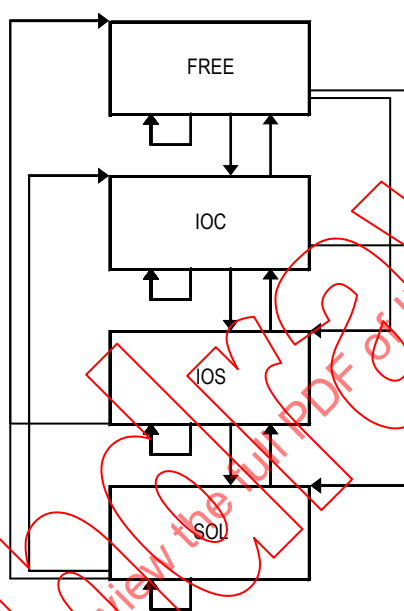
The local service primitives including their associated parameters issued or received by OWNSM are described in the service definition and shown in Table 311.

Table 311 – Local primitives issued or received by OWNSM

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.4.6.2.2.3 State transition diagram**7.4.6.2.2.3.1 OWNSM**

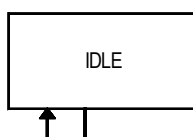
The state transition diagram is shown in Figure 51.

**Figure 53 – State transition diagram OWNSM****States of the OWNSM**

FREE	The submodule has no owner
SOL	The submodule is superordinated locked
IOS	The submodule is owned by an IO supervisor
IOC	The submodule is owned by an IO controller

7.4.6.2.2.3.2 ASSSM

The state transition diagram is shown in Figure 54.

**Figure 54 – State transition diagram ASSSM****States of the ASSSM**

IDLE	The submodule is waiting for an association

7.4.6.2.2.4 State machine description

7.4.6.2.2.4.1 OWNSM

The state machine treats the ownership of a particular submodule. The handover of the ownership to an IO supervisor or to the application (superordinated locked) is accompanied by the Controlled By Supervisor alarm, the release by the Released alarm.

If the submodule is not usable by the next one in the ownership queue due to being the wrong submodule (WRONG) plugged as waited for or having no submodule plugged at the subslot (NOSUBMODULE). In this case no Released alarm is sent to the new owner as an optimization (“hidden ownership change”), because the submodule is still not usable for him. Note however that all further alarms regarding this submodule (e.g. a Plug alarm) are sent to the new owner.

Thus an IO Controller implementation shall accept alarms for submodules even for those he is not owning from his current point of view.

The state machine handles the ownership queue (i.e. those application relations waiting in line for the particular submodule) and the flag SupervisorTakeoverAllowed.

It additionally filters write requests, if issued via a different application relation than the current owner.

7.4.6.2.2.4.2 ASSSM

The state machine treats the association of a particular submodule.

7.4.6.2.2.5 State table

Table 312 defines the state table for the selected behavior of the OWNSM. This state table shall be used if Submodule Properties.Shared Input is set to 0x00.

Table 313 defines the state table for the selected behavior of the ASSSM. This state table shall be used if Submodule Properties.Shared Input is set to 0x01.

Table 312 – State table OWNSM

#	Current State	Event /Condition =>Action	Next State
1	FREE	Connect.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	FREE
2	FREE	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType != IOSAR => AddToOwnershipAdministration ()	IOC
3	FREE	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType == IOSAR => AddToOwnershipAdministration ()	IOS
4	FREE	UserSetSMState.req () /Command == SOL =>	SOL

#	Current State	Event /Condition =>Action	Next State
		ignore UserSetSMState.cnf ()	
5	FREE	UserSetSMState.req () /Command != SOL => ignore UserSetSMState.cnf ()	FREE
6	FREE	Release.req () => ignore	FREE
7	FREE	ReleasedAlarm.cnf () => ignore	FREE
8	FREE	ControlledAlarm.cnf () => ignore	FREE
9	FREE	Write.ind () => ErrorDecode := PNORW ErrorCode1 := access denied Write.rsp (-)	FREE
10	FREE	Read.ind () => Fetch data for read response Read.rsp () //NOTE Depending whether the user is able to return data (+) or (-) is returned	FREE
11	IOC	Connect.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	IOC
12	IOC	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType != IOSAR => AddToOwnershipAdministration ()	IOC
13	IOC	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType == IOSAR && Owner.ARProperties.SupervisorTakeOverAllowed == TRUE => AddToOwnershipAdministration () Create ControlledBySupervisor alarm PDU to be sent to the now previous owner ControlledAlarm.req ()	IOS
14	IOC	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType == IOSAR && Owner.ARProperties.SupervisorTakeOverAllowed == FALSE => AddToOwnershipAdministration ()	IOC
15	IOC	UserSetSMState.req () /Command == SOL => AddToOwnershipAdministration () Create ControlledBySupervisor alarm PDU to be sent to the now previous owner ControlledAlarm.req ()	SOL

#	Current State	Event /Condition =>Action	Next State
		UserSetSMState.cnf ()	
16	IOC	UserSetSMState.req () /Command != SOL => ignore UserSetSMState.cnf ()	IOC
17	IOC	Release.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	IOC
18	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == FREE => ignore	FREE
19	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOC && (UserCheckIdent () == WRONG UserCheckIdent () == NOSUBMODULE) => ignore //NOTE No Released alarm is sent, because the submodule is still not usable for the new owner.	IOC
20	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOC && (UserCheckIdent () == OK UserCheckIdent () == SUBSTITUTE) => Create Released alarm PDU to be sent to the new owner ReleasedAlarm.req ()	IOC
21	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOS && (UserCheckIdent () == WRONG UserCheckIdent () == NOSUBMODULE) //NOTE IOS was blocked by TakeoverNotAllowed => ignore //NOTE No Released alarm is sent, because the submodule is still not usable for the new owner.	IOS
22	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOS && (UserCheckIdent () == OK UserCheckIdent () == SUBSTITUTE) //NOTE IOS was blocked by TakeoverNotAllowed => Create Released alarm PDU to be sent to the new owner ReleasedAlarm.req ()	IOS
23	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOC, ARset => ignore //NOTE SystemRedundancy case	IOC

#	Current State	Event /Condition =>Action	Next State
24	IOC	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == UNCHANGED => ignore	IOC
25	IOC	ReleasedAlarm.cnf () => ignore	IOC
26	IOC	ControlledAlarm.cnf () => ignore	IOC
27	IOC	Write.ind () /AREP == GetCurrentOwner () //NOTE Accept write only from the current owner => Give data for write to user and fetch status for write response Write.rsp () //NOTE Depending whether the user is able to handle data (+) or (-) is returned	IOC
28	IOC	Write.ind () /AREP != GetCurrentOwner () //NOTE Accept write only from the current owner => ErrorDecode := PNORW ErrorCode1 := access denied Write.rsp (-)	IOC
29	IOC	Read.ind () => Fetch data for read response Read.rsp () //NOTE Depending whether the user is able to return data (+) or (-) is returned	IOC
30	IOS	Connect.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	IOS
31	IOS	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType != IOSAR => AddToOwnershipAdministration ()	IOS
32	IOS	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock && ARType == IOSAR => AddToOwnershipAdministration ()	IOS
33	IOS	UserSetSMState.req () /Command == SOL => AddToOwnershipAdministration () Create ControlledBySupervisor alarm PDU to be sent to the now previous owner ControlledAlarm.req () UserSetSMState.cnf ()	SOL
34	IOS	UserSetSMState.req () /Command != SOL => ignore UserSetSMState.cnf ()	IOS

#	Current State	Event /Condition =>Action	Next State
35	IOS	Release.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	IOS
36	IOS	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == FREE => ignore	FREE
37	IOS	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOC && (UserCheckIdent () == WRONG UserCheckIdent () == NOSUBMODULE) => ignore //NOTE No Released alarm is sent, because the submodule is still not usable for the new owner.	IOC
38	IOS	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOC && (UserCheckIdent () == OK UserCheckIdent () == SUBSTITUTE) => Create Released alarm PDU to be sent to the new owner ReleasedAlarm.req ()	IOC
39	IOS	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOS && (UserCheckIdent () == WRONG UserCheckIdent () == NOSUBMODULE) => ignore //NOTE No Released alarm is sent, because the submodule is still not usable for the new owner.	IOS
40	IOS	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == IOS && (UserCheckIdent () == OK UserCheckIdent () == SUBSTITUTE) => Create Released alarm PDU to be sent to the new owner ReleasedAlarm.req ()	IOS
41	IOS	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock && RemoveAndGetOwnerFromAdministration () == UNCHANGED => ignore	IOS
42	IOS	ReleasedAlarm.cnf () => ignore	IOS
43	IOS	ControlledAlarm.cnf () => ignore	IOS
44	IOS	Write.ind () /AREP == GetCurrentOwner () //NOTE Accept write only from the current owner =>	IOS

#	Current State	Event /Condition =>Action	Next State
		Give data for write to user and fetch status for write response Write.rsp () //NOTE Depending whether the user is able to handle data (+) or (-) is returned	
45	IOS	Write.ind () /AREP != GetCurrentOwner () //NOTE Accept write only from the current owner => ErrorDecode := PNORW ErrorCode1 := access denied Write.rsp (-)	IOS
46	IOS	Read.ind () => Fetch data for read response Read.rsp () //NOTE Depending whether the user is able to return data (+) or (-) is returned	IOS
47	SOL	Connect.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	SOL
48	SOL	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock => AddToOwnershipAdministration ()	SOL
49	SOL	UserSetSMState.req () /Command == SOL => ignore UserSetSMState.cnf ()	SOL
50	SOL	UserSetSMState.req () /Command != SOL && RemoveAndGetOwnerFromAdministration () == FREE => ignore UserSetSMState.cnf ()	FREE
51	SOL	UserSetSMState.req () /Command != SOL && RemoveAndGetOwnerFromAdministration () == IOC && (UserCheckIdent () == WRONG UserCheckIdent () == NOSUBMODULE) => //NOTE No Released alarm is sent, because the submodule is still not usable for the new owner. UserSetSMState.cnf ()	IOC
52	SOL	UserSetSMState.req () /Command != SOL && RemoveAndGetOwnerFromAdministration () == IOC && (UserCheckIdent () == OK UserCheckIdent () == SUBSTITUTE) => Create Released alarm PDU to be sent to the new owner ReleasedAlarm.req () UserSetSMState.cnf ()	IOC
53	SOL	UserSetSMState.req () /Command != SOL && RemoveAndGetOwnerFromAdministration () == IOS && (UserCheckIdent () == WRONG UserCheckIdent () == NOSUBMODULE) =>	IOS

#	Current State	Event /Condition =>Action	Next State
		//NOTE No Released alarm is sent, because the submodule is still not usable for the new owner. UserSetSMState.cnf ()	
54	SOL	UserSetSMState.req () /Command != SOL && RemoveAndGetOwnerFromAdministration () == IOS && (UserCheckIdent () == OK UserCheckIdent () == SUBSTITUTE) => Create Released alarm PDU to be sent to the new owner ReleasedAlarm.req () UserSetSMState.cnf ()	IOS
55	SOL	Release.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	SOL
56	SOL	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock => RemoveAndGetOwnerFromAdministration () //NOTE In this case only the ownership administration is updated	SOL
57	SOL	ReleasedAlarm.cnf () => ignore	SOL
58	SOL	ControlledAlarm.cnf () => ignore	SOL
59	SOL	Write.ind () => ErrorDecode := PNORW ErrorCode1 := access denied Write.rsp (-)	SOL
60	SOL	Read.ind () => Fetch data for read response Read.rsp () //NOTE Depending whether the user is able to return data (+) or (-) is returned	SOL

Table 313 – State table ASSSM

#	Current State	Event /Condition =>Action	Next State
1	IDLE	Connect.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	IDLE
2	IDLE	Connect.req () /Submodule is in list of the ExpectedSubmoduleBlock => AddToAssociation ()	IDLE
3	IDLE	Release.req () /Submodule is not in list of the ExpectedSubmoduleBlock => ignore	IDLE
4	IDLE	Release.req () /Submodule is in list of the ExpectedSubmoduleBlock => RemoveFromAssociation ()	IDLE

#	Current State	Event /Condition =>Action	Next State
5	IDLE	Write.ind () => ErrorDecode := PNORW ErrorCode1 := access denied Write.rsp (-)	IDLE
6	IDLE	Read.ind () => Fetch data for read response Read.rsp () //NOTE Depending whether the user is able to return data (+) or (-) is returned	IDLE

7.4.6.2.2.6 Functions, Macros, Timers and Variables

Table 314 contains the functions, macros, timers and variables used by OWNSM and their arguments and their descriptions.

Table 314 – Functions, Macros, Timers and Variables used by OWNSM

Name	Type	Function/Meaning
RemoveAndGetOwnerFromAdministration	Function	This local function removes the given AREP from the ownership administration. It returns whether/where to the ownership changed and the AREP of the application relation now owning the submodule. The ownership is primary handled by the first come first served principle. This means that the first connect.req (first AR) requesting a "free" submodule will get the ownership. The exceptions to this principle are higher priority ownership request by the IO supervisor or local means (Superordinated lock) or by the TakeoverNotAllowed property of a IOC-AR. Principle: A supervisor may take away a submodule from an IO controller and local means may take away a submodule from an IO supervisor or IO controller. FREE This submodule has no longer an owner IOC The ownership of the submodule is transferred to an IOC IOC, ARset The ownership of the submodule is transferred to an IOC within the same ARset as the previous owner IOS The ownership of the submodule is transferred to an IOS UNCHANGED The ownership of the submodule did not change; i.e. the AREP removed was not the current owner
GetCurrentOwner	Function	This local function returns the AREP of the submodule's current owner
AddToOwnershipAdministration	Function	This local function adds the new owner or a candidate, addressed by the AREP, to the ownership administration. Superordinated lock is treated like an AR.

Name	Type	Function/Meaning
UserCheckIdent	Function	This local function asks the User to check the expected ModuleID and SubmoduleID against the real one. Returns the ident information for the expected submodule: OK: is returned, if a submodule is plugged at the specified location (API, slot and subslot) and it's the right one; i.e. its real ident information (RealModuleID, RealSubmoduleID) is equivalent to the expected ident information (ExpModuleID and ExpSubmoduleID). SUBSTITUTE: is returned, if a submodule is plugged at the specified location (API, slot and subslot) and is able to be a decent substitute; i.e. its real ident information (RealModuleID, RealSubmoduleID) is able to substitute the expected ident information (ExpModuleID and ExpSubmoduleID). WRONG: is returned, if a submodule is plugged at the specified location (API, slot and subslot) and it's wrong; i.e. if its real ident information (RealModuleID, RealSubmoduleID) does not match the expected ident information (ExpModuleID and ExpSubmoduleID), or if it doesn't support the expected API. NOSUBMODULE: is returned, if no submodule is plugged at the location specified by API, slot and subslot.

Table 315 – Functions, Macros, Timers and Variables used by ASSSM

Name	Type	Function/Meaning
AddToAssociation	Function	This local function adds the new associate, addressed by the AREP, to the association. Adding a submodule to the association shall not create an entry in the module diff block.
RemoveFromAssociation	Function	This local function removes the new associate, addressed by the AREP, to the association.

7.4.6.2.2.7 Truth Table

Table 316 contains the truth table used by OWNSM.

Table 316 – Rules for Submodule State.AR Info

Input		Output
Current Owner	Parameter Check Pending	Submodule State.AR Info
OWN Owner is in same AR Set	No	OWN
OWN Owner is in same AR Set	Yes	APPLICATION_READY_PENDING
Superordinated Locked	-	SUPERORDINATED_LOCKED
Other IO Controller	-	LOCKED_BY_IO_CONTROLLER
Other IO Supervisor	-	LOCKED_BY_IO_SUPERVISOR

7.4.6.2.3 Plug

7.4.6.2.3.1 Primitive definitions

7.4.6.2.3.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by plug state machine (PLUGSM) are described in the service definition and shown in Table 317.

Table 317 – Remote primitives issued or received by PLUGSM

Primitive	Source	Destination	Associated parameters	Functions
Alarm.cnf ()	FAL	PLUGSM	see 7.3.5.3	—
ApplicationReady.cnf ()	FAL	PLUGSM	see 7.3.6.3.7	—
PrmEnd.ind	FAL	PLUGSM	see 7.3.6.3.6	—
Write.ind	FAL	PLUGSM	see 7.3.1.4.2.1	—
ReturnOfSubmodulAlarm.req	local	PLUGSM	AREP, API, Slot, Subslot, UserInfo;	—
UserControl.rsp ()	local	PLUGSM	AREP, PrmEnd	—
UserWriteRecord.rsp ()	local	PLUGSM		—
ReleasedAlarm.req	OWNSM	PLUGSM	AREP, API, Slot, Subslot, UserInfo;	—
Alarm.req	PLUGSM	FAL	see 7.3.5.3	—
ApplicationReady.req	PLUGSM	FAL	see 7.3.6.3.7	—
PrmEnd.rsp ()	PLUGSM	FAL	see 7.3.6.3.6	—
Write.rsp ()	PLUGSM	FAL	see 7.3.1.4.2.1	—
ReturnOfSubmodulAlarm.cnf ()	PLUGSM	local	AREP, API, Slot, Subslot, UserInfo;	—
UserControl.ind	PLUGSM	local	AREP, PrmEnd	—
UserWriteRecord.ind	PLUGSM	local		—
ReleasedAlarm.cnf ()	PLUGSM	OWNSM	AREP, API, Slot, Subslot, UserInfo;	—
PlugAlarm.cnf ()	PLUGSM	RSMSM		—
PlugWrongAlarm.cnf ()	PLUGSM	RSMSM	AREP, API, Slot, Subslot, UserInfo;	—
PlugAlarm.req	RSMSM	PLUGSM	AREP, API, Slot, Subslot, UserInfo;	—

Primitive	Source	Destination	Associated parameters	Functions
PlugWrongAlarm.req	RSMSM	PLUGSM	AREP, API, Slot, Subslot, UserInfo;	—

7.4.6.2.3.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by PLUGSM are described in the service definition and shown in Table 318.

Table 318 – Local primitives issued or received by PLUGSM

Primitive	Source	Destination	Associated parameters	Functions
Local New Output_ind	FAL	PLUGSM	see 7.3.2.5.14	—
Local Set Input_req	PLUGSM	FAL	see 7.3.2.5.1	—
Local Set Output IOCS_req	PLUGSM	FAL	see 7.3.2.5.11	—

7.4.6.2.3.2 State transition diagram

The state transition diagram is shown in Figure 55.

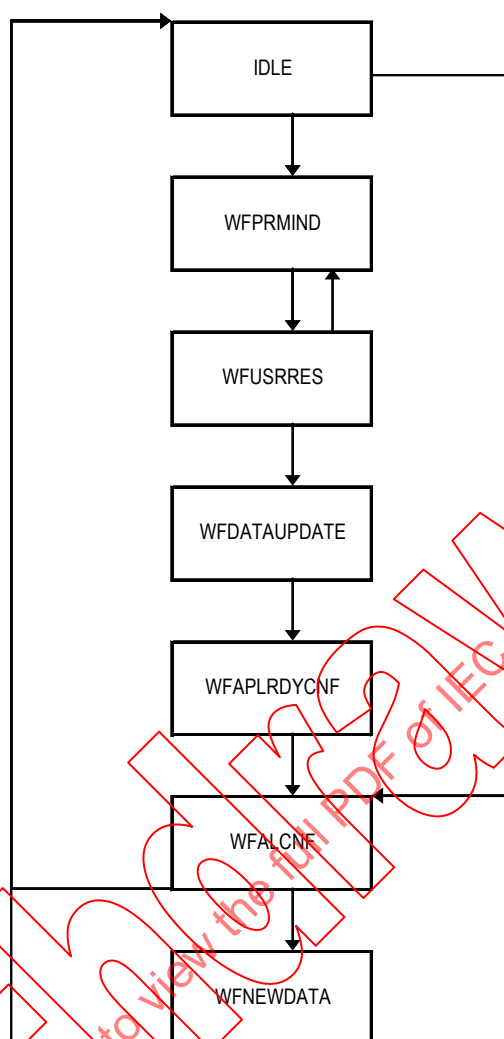


Figure 55 – State transition diagram PLUGSM

States of PLUGSM

IDLE	Idle state
WFPRMIND	Wait for PrmEnd indication
WFUSRRES	Wait for User response
WFDATAUPDATE	Wait for Data Update
WFAPLRDYCNF	Wait for Appl_Rdy confirmation
WFALCNF	Wait for Alarm confirmation
WFNEWDATA	Wait for New Data indication

7.4.6.2.3.3 State machine description

This state machine handles a plug of a submodule including the re-parameterization of the submodule. The plug sequence is initiated by plugging a submodule (including a wrong one), by ownership change or else by return of a submodule (transition from BAD to GOOD).

7.4.6.2.3.4 State table

Table 319 defines the state table for the plug behavior.

Table 319 – State table PLUGSM

#	Current State	Event /Condition =>Action	Next State
1	IDLE	PlugAlarm.req () => TYPE = Plug Alarm.req ()	WFPRMIND
2	IDLE	ReleasedAlarm.req () => TYPE = Released Alarm.req ()	WFPRMIND
3	IDLE	PlugWrongAlarm.req () => TYPE = Wrong Alarm.req ()	WFALCNF
4	IDLE	ReturnOfSubmodulAlarm.req () => TYPE = Return Alarm.req ()	WFALCNF
5	WFPRMIND	Write.ind () => UserWriteRecord.ind ()	WFUSRRES
6	WFPRMIND	PrmEnd.ind () => UserControl.ind ()	WFUSRRES
7	WFUSRRES	UserWriteRecord.rsp () => Write.rsp ()	WFPRMIND
8	WFUSRRES	UserControl.rsp () => PrmEnd.rsp ()	WFDATAUPDATE
9	WFDATAUPDATE	UserUpdate () => Local Set Input.req () Local Set Output IOCS.req () ApplicationReady.req ()	WFAPLRDYCNF
10	WFAPLRDYCNF	ApplicationReady.cnf () =>	WFALCNF
11	WFALCNF	Alarm.cnf () /((TYPE = Plug) (TYPE = Released)) && (PNIOSTATUS = OK) => Reset NewData Flag	WFNEWDATA
12	WFALCNF	Alarm.cnf () /((TYPE = Wrong) && (PNIOSTATUS = OK)) => PlugWrongAlarm.cnf (+)	IDLE
13	WFALCNF	Alarm.cnf () /((TYPE = Return) && (PNIOSTATUS = OK)) => ReturnOfSubmodulAlarm.cnf (+)	IDLE
14	WFALCNF	Alarm.cnf () /((TYPE = Plug) && (PNIOSTATUS <> OK)) => PlugAlarm.cnf (-)	IDLE
15	WFALCNF	Alarm.cnf () /((TYPE = Released) && (PNIOSTATUS <> OK)) => ReleasedAlarm.cnf (-)	IDLE
16	WFALCNF	Alarm.cnf () /((TYPE = Wrong) && (PNIOSTATUS <> OK)) => PlugWrongAlarm.cnf (-)	IDLE

#	Current State	Event /Condition =>Action	Next State
17	WFALCNF	Alarm.cnf () /(TYPE = Return) && (PNIOSTATUS <> OK) => ReturnOfSubmodulAlarm.cnf (-)	IDLE
18	WFNEWDATA	Local New Output_ind () /TYPE = Plug => PlugAlarm.cnf (+)	IDLE
19	WFNEWDATA	Local New Output_ind () /TYPE = Released => ReleasedAlarm.cnf (+)	IDLE

7.4.6.2.3.5 Functions, Macros, Timers and Variables

Table 320 contains the functions, macros, timers and variables used by PLUGSM and their arguments and their descriptions.

Table 320 – Functions, Macros, Timers and Variables used by PLUGSM

Name	Type	Function/Meaning
UserUpdate	Function	This local function is an acknowledge from the user to use the data and issue an application ready. It uses the following parameters: AREP, API, Slot, Subslot, Data, IOXS

7.4.6.2.4 Pull

7.4.6.2.4.1 Primitive definitions

7.4.6.2.4.1.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by pull state machine (PULLSM) are described in the service definition and shown in Table 321.

Table 321 – Remote primitives issued or received by PULLSM

Primitive	Source	Destination	Associated parameters	Functions
Alarm.cnf ()	FAL	PULLSM	see 7.3.5.3	—
ControlledAlarm.req	OWNSM	PULLSM	AREP, API, Slot, Subslot, UserInfo;	—
Alarm.req	PULLSM	FAL	see 7.3.5.3	—
ControlledAlarm.cnf ()	PULLSM	OWNSM	AREP, API, Slot, Subslot, UserInfo;	—

Primitive	Source	Destination	Associated parameters	Functions
PullAlarm.cnf ()	PULLSM	RSMSM	AREP, API, Slot, Subslot, UserInfo;	—
PullAlarm.req	RSMSM	PULLSM	AREP, API, Slot, Subslot, UserInfo;	—

7.4.6.2.4.1.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by PULLSM are described in the service definition and shown in Table 322.

Table 322 – Local primitives issued or received by PULLSM

Primitive	Source	Destination	Associated parameters	Functions
—	—	—	—	—

7.4.6.2.4.2 State transition diagram

The state transition diagram is shown in Figure 56.

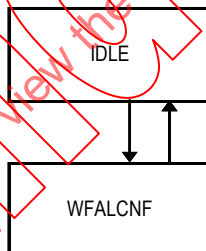


Figure 56 – State transition diagram PULLSM

States of PULLSM

IDLE	Idle state
WFALCNF	Wait for Alarm confirmation

7.4.6.2.4.3 State machine description

This state machine handles a pull of a submodule. The pull sequence is initiated by pulling a submodule or by ownership change.

7.4.6.2.4.4 State table

Table 323 defines the state table PULLSM.

Table 323 – State table PULLSM

#	Current State	Event /Condition =>Action	Next State
1	IDLE	PullAlarm.req () => TYPE = Pull Alarm.req ()	WFALCNF
2	IDLE	ControlledAlarm.req () => TYPE= Controlled Alarm.req ()	WFALCNF
4	WFALCNF	Alarm.cnf () /(TYPE = Pull) && (PNIOSTATUS = OK) => PullAlarm.cnf (+)	IDLE
5	WFALCNF	Alarm.cnf () /(TYPE = Controlled) && (PNIOSTATUS = OK) => ControlledAlarm.cnf (+)	IDLE
6	WFALCNF	Alarm.cnf () /(TYPE = Pull) && (PNIOSTATUS <> OK) => PullAlarm.cnf(-)	IDLE
7	WFALCNF	Alarm.cnf () /(TYPE = Controlled) && (PNIOSTATUS <> OK) => ControlledAlarm.cnf(-)	IDLE

7.4.6.2.4.5 Functions, Macros, Timers and Variables

Table 324 contains the functions, macros, timers and variables used by PULLSM and their arguments and their descriptions.

Table 324 – Functions, Macros, Timers and Variables used by PULLSM

Name	Type	Function/Meaning
	—	—

7.4.7 Behavior for PTCP within IO devices

7.4.7.1 Diagnosis and Alarms

PTCP synchronisation events will cause diagnosis events and alarm notifications. Both will only be generated if an AR is established and a proper sync record is present.

7.4.7.2 Sync error dependencies

There are some dependencies of sync errors, so not all errors occur at once.

If no synchronization mismatch is present we are synchronized. “Wrong PTCPSubDomainID” and “Wrong IRDataID” requires a sync message to be received. “Jitter out of Boundary” requires a sync message to be received and correct PTCPSubDomainID and IRDataID.

These dependencies are handled within PTCP “Local Sync State Info” service.

7.4.7.3 Primitive definitions

7.4.7.3.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by synchronisation diagnosis (SYNC_DIAG) devices are described in the service definition and shown in Table 325.

Table 325 – Remote primitives issued or received by SYNC_DIAG

Primitive	Source	Destination	Associated parameters	Functions
Local Diagnosis Event.req	SYNC_DIAG	FAL	see 7.3.4.3.5	—
Alarm.req	SYNC_DIAG	FAL	see 7.3.5.3	—
Alarm.cnf ()	FAL	SYNC_DIAG	see 7.3.5.3	—

7.4.7.3.2 Primitives exchanged between local machines

The local service primitives including their associated parameters issued or received by SYNC_DIAG are described in the service definition and shown in Table 326.

Table 326 – Local primitives issued or received by SYNC_DIAG

Primitive	Source	Destination	Associated parameters	Functions
Sync State Change.ind	local	SYNC_DIAG	see 6.3.2.3.7	—

7.4.7.4 State transition diagram

The state transition diagram of the SYNC_DIAG is shown in Figure 57

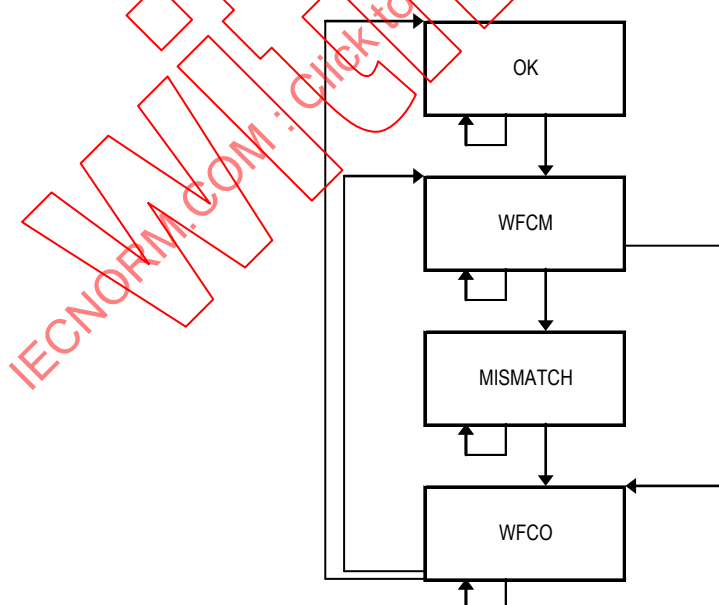


Figure 57 – State transition diagram SYNC_DIAG

States of the SYNC_DIAG

OK	Initiale state and synchronization ok
WFCM	Wait for alarm confirm while going to

	mismatch
MISMATCH	synchronization diagnosis exists
WFCO	Wait for alarm confirm while going to ok

7.4.7.5 State machine description

This state machine cares about the diagnosis issues regarding synchronization. It is triggered by sync state changed indications and issues coming and going alarms.

7.4.7.6 State table

Table 327 defines the state table for SYNC_DIAG.

Table 327 – State table SYNC_DIAG

#	Current State	Event /Condition => Action	Next State
1	OK	Sync State Change.ind () /SyncState == SYNCHRONIZED => ignore	OK
2	OK	Sync State Change.ind () /SyncState == NOT_SYNCHRONIZED => SYNC_MISMATCH (SyncID, ErrorType) NextState := MISMATCH Alarm.req ()	WFCM
3	OK	Alarm.cnf () => ignore	OK
4	WFCM	Sync State Change.ind () /SyncState == SYNCHRONIZED => NextState := OK	WFCM
5	WFCM	Sync State Change.ind () /SyncState == NOT_SYNCHRONIZED => NextState := MISMATCH	WFCM
6	WFCM	Alarm.cnf () /NextState == OK => SYNC_OK (SyncID) Alarm.req ()	WFCO
7	WFCM	Alarm.cnf () /NextState == MISMATCH => ignore	MISMATCH
8	MISMATCH	Alarm.cnf () => ignore	MISMATCH
9	MISMATCH	Sync State Change.ind () /SyncState == SYNCHRONIZED => SYNC_OK (SyncID) NextState := OK Alarm.req ()	WFCO
10	MISMATCH	Sync State Change.ind () /SyncState == NOT_SYNCHRONIZED => ignore	MISMATCH

#	Current State	Event /Condition =>Action	Next State
11	WFCO	Sync State Change.ind () /SyncState == SYNCHRONIZED => NextState := OK	WFCO
12	WFCO	Sync State Change.ind () /SyncState == NOT_SYNCHRONIZED => NextState := MISMATCH	WFCO
13	WFCO	Alarm.cnf () /NextState == MISMATCH => SYNC_MISMATCH (SyncID, ErrorType) Alarm.req ()	WFCM
14	WFCO	Alarm.cnf () /NextState == OK => ignore	OK

7.4.7.7 Functions, Macros, Timers and Variables

Table 328 contains the functions, macros, timers and variables used by the state table of SYNC_DIAG, their arguments and their descriptions.

Table 328 – Functions, Macros, Timers and Variables used by SYNC_DIAG

Name	Type	Function/Meaning
SYNC_OK(SyncID)	Function	No Sync mismatch. Sync OK. Alarm Priority := ALARM_HIGH or ALARM_LOW if (SyncID == 0) Alarm type:= Sync Data Change Notification else Alarm type:= Time data changed notification Alarm Item.User Structure Identifier := 0x8000 (ChannelDiagnosisData) Channel Number := 0x8000 Channel Properties.type := 0 Channel Properties.Accumulative := 0 Channel Properties.MaintenanceRequired := 0 (Diagnosis) Channel Properties.Demanded := 0 Channel Properties.Specifier := 0 (all disappear) Channel Properties.Direction := 0 if (SyncID == 0) Channel Error type := Sync mismatch else Channel Error type := Time mismatch Local Diagnosis Event_ind ()

Name	Type	Function/Meaning
SYNC_MISMATCH(SyncID,ErrorType)	Function	Sync mismatch. Issue an Alarm Notification and Diagnosis Event. Only changes of error states will cause an Alarm Notification and Diagnosis Event! Alarm Priority:= ALARM_HIGH or ALARM_LOW if (SyncID == 0) Alarm type:= Sync Data Change Notification else Alarm type:= Time data changed notification Alarm Item.User Structure Identifier := 0x8002 (ExtChannelDiagnosisData) Channel Number := 0x8000 Channel Properties.type := 0 Channel Properties.Accumulative := 0 Channel Properties.MaintenanceRequired := 1 Channel Properties.Demanded := 0 Channel Properties.Specifier := Appears Channel Properties.Direction := 0 if (SyncID == 0) Channel Error type := Sync mismatch else Channel Error type := Time mismatch Ext Channel Error type := ErrorType Local Diagnosis Event_ind ()
NextState	Variable	This local variable contains the desired state

7.5 Behavior of an IO controller

7.5.1 Overview

Each IO Controller or IO Supervisor shall have a startup machine for establishing an AR.

7.5.2 Behavior of an IO controller during startup

7.5.2.1 General

This subclause describes the client application behavior of an IO controller during startup (CTLISM) of the user application as an example.

NOTE This subclause provides selected behavior only and is not a comprehensive definition of the client behavior.

In the startup phase data records shall be written in the following transfer sequence:

- AR related data records shall be transferred first in ascending order according to their index.
- Within the IO Device, all APIs are processed in ascending order.
- Within each API, all slots carrying a module are processed in ascending order.
- Within each module, all subslots carrying a submodule are processed in ascending order.
- Within each submodule,
 - data records defined by IEC 61158-6-10 (having an index $\geq$ 0x8000) will always be transferred first in ascending order according to their index.
 - manufacturer specific data records (having an index $<$ 0x8000) will follow afterwards
 - a) according to their attribute @TransferSequence as given in the device's GSD,

- or if the attribute @TransferSequence is not stated, in ascending order according to their index.

7.5.2.2 Primitive definitions

7.5.2.2.1 Primitives exchanged between remote machines

The service primitives including their associated parameters issued or received by CTLSM are described in the service definition and shown in Table 329.

Table 329 – Remote primitives issued or received by CTLSM

Primitive	Source	Destination	Associated parameters	Functions
ApplicationReady.ind	FAL	CTLSM	see 7.3.6.3.7	—
Connect.cnf (+)	FAL	CTLSM	see 7.3.6.3.1.2	—
Prm End.cnf	FAL	CTLSM	see 7.3.6.3.6	—
Write.cnf	FAL	CTLSM	see 7.3.1.4.2.1	—
ApplicationReady.rsp ()	CTLSM	FAL	see 7.3.6.3.7	—
Connect.req	CTLSM	FAL	see 7.3.6.3.1.2	—
Prm End.req	CTLSM	FAL	see 7.3.6.3.6	—
Write.req	CTLSM	FAL	see 7.3.1.4.2.1	—

7.5.2.2.2 Primitives exchanged between local machines

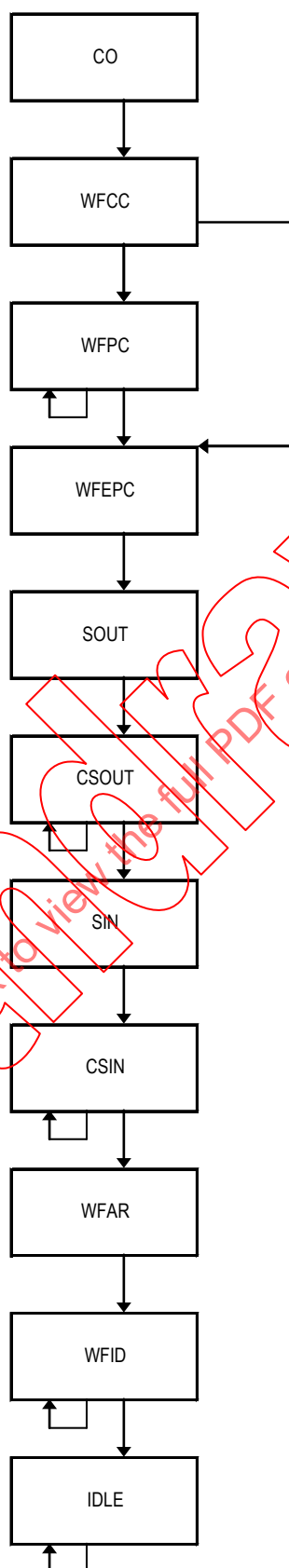
The local service primitives including their associated parameters issued or received by CTLSM are described in the service definition and shown in Table 330.

Table 330 – Local primitives issued or received by CTLSM

Primitive	Source	Destination	Associated parameters	Functions
Local Set Input IOCS.cnf (+)	FAL	CTLSM	see 7.3.2.5.2	—
Local Set Output.cnf (+)	FAL	CTLSM	see 7.3.2.5.10	—
Local Set Input IOCS.req	CTLSM	FAL	see 7.3.2.5.2	—
Local Set Output.req	CTLSM	FAL	see 7.3.2.5.10	—
Local New Input.ind	FAL	CTLSM	see 7.3.2.5.5	—

7.5.2.3 State transition diagram

Figure 58 shows the state diagram for the startup.



States of CTLSM

CO	Initiate the connect request
WFCC	Wait for a connect response
WFPC	Wait for a write confirmation
WFEPc	Wait for Prm End confirmation
SOUT	Initialize output according parameterization
CSOUT	Wait for output initialization
SIN	Initialize input buffer according to the real inputs
CSIN	Wait for input buffer initialization
WFAR	Wait for ApplReady indication
WFID	Wait for IN Data
IDLE	Working state IN Data

7.5.2.4 State machine description

This state machine describes the application behavior of an IO controller during startup of a particular application relation to an IO device.

7.5.2.5 State table

Table 331 defines the state table for IO controller during startup.

Table 331 – State table CTLsm

#	Current State	Event /Condition => Action	Next State
1	CO	=> SetConnectRequestData("ExpectedConfiguration") AR Parameter Block.SessionKey:=StoredSessionKey+1 StoredSessionKey:=StoredSessionKey+1 Connect.req()	WFCC
2	WFCC	Connect.cnf (+) /CheckResponseBlocks()!=Ok && ParameterToSend()!=Ok => PrepareWritePDU() Write.req()	WFPC
3	WFCC	Connect.cnf (+) /CheckResponseBlocks()!=Ok && ParameterToSend()!=Ok => Prm End.req()	WFEPc
4	WFCC	Connect.cnf (+) /CheckResponseBlocks()!=Ok && ParameterToSend()!=Ok => AdaptConfiguration() Write.req()	WFPC
5	WFCC	Connect.cnf (+) /CheckResponseBlocks()!=Ok && ParameterToSend()!=Ok => AdaptConfiguration() Prm End.req()	WFEPc
6	WFPC	Write.cnf () /ParameterToSend()!=Ok => Session Key:=StoredSessionKey PrepareWritePDU() Write.req()	WFPC

#	Current State	Event /Condition =>Action	Next State
7	WFPC	Write.cnf () /ParameterToSend() != Ok => Session Key = StoredSessionKey Prm End.req ()	WFPC
8	WFPC	Prm End.cnf () =>	SOUT
9	SOUT	=> For all involved slot/subslots do: Slot.Subslot.IOCS := GOOD Subslot Output Data := Slot.Subslot.Initial Value For all involved slot/subslots do: Local Set Output_req ()	CSOUT
10	CSOUT	Local Set Output.cnf (+) /not last one =>	CSOUT
11	CSOUT	Local Set Output.cnf (+) /last one =>	SIN
12	SIN	=> For all involved slot/subslots match expected configuration do: Slot := Slot Number; Subslot := Subslot Number IOCS := GOOD For all involved slot/subslots does not match expected configuration do: Slot := Slot Number; Subslot := Subslot Number IOCS := BAD_BY_CONTROLLER For all involved slot/subslots do: Local Set Input IOCS_req ()	CSIN
13	CSIN	Local Set Input IOCS.cnf (+) /not last one =>	CSIN
14	CSIN	Local Set Input IOCS.cnf (+) /last one =>	WFAR
15	WFAR	ApplicationReady.ind () /Session Key = StoredSessionKey => ApplicationReady.rsp ()	WFID
16	WFID	Local New Input_ind () /not last one =>	WFID
17	WFID	Local New Input_ind () /last one => SumARinData ()	IDLE
18	IDLE	=> ignore	IDLE

7.5.2.6 Functions, Macros, Timers and Variables

Table 332 contains the functions, macros, timers and variables used by CTLSM and their arguments and their descriptions.

Table 332 – Functions, Macros, Timers and Variables used by CTLSM

Name	Type	Function/Meaning
CheckResponseBlocks	Function	This local function checks whether the syntax and the semantics of the response is correct
ParameterToSend	Function	This local function indicates the existence of any parameter record
AdaptConfiguration	Function	This local function adapts the list of parameter records according to the connect response
SumARinData	Function	This local function is issued from DEVSM if all consumer reaches the state Data
StoredSessionKey	Variable	This local variable stores the session key from the connect service
SetConnectRequestData	Function	This local function configures the connect request
PrepareWritePDU	Function	This local function fetches the parameter from the local data base and constructs the write multiple PDU

7.6 Application characteristics

7.6.1 Device ident number

The Device Ident Number is necessary for all IO devices. For each type of device an individual Device Ident Number will be needed. This Device Ident Number is not a series number. If a manufacturer has a Device Ident Number for one type of device it is allowed to use this number for all produced devices of the same type. At devices of the same type but with differences in the amount of inputs and outputs it is possible to use the same Device Ident Number for each produced IO Device.

Each device type shall have an individual Device Ident Number which is part of the object UUID.

The Device Ident Number consists of a Vendor ID, which is a globally organizational administered number, and a Device ID, which is administered by the manufacturer.

7.6.2 Network topology

IO controller and IO devices make provisions concerning the network connection. The DTEs of such devices shall support a bit rate of 100 MBit/s in full duplex mode. However, some topologies include e.g. wireless segments which do not fulfill these requirements. In such case, the wireless part shall be connected in such a way that only application relationships that use these network path suffer under lower performance. The adaptation can be done by means of the attribute reduction ratio. An example is depicted in Figure 59.

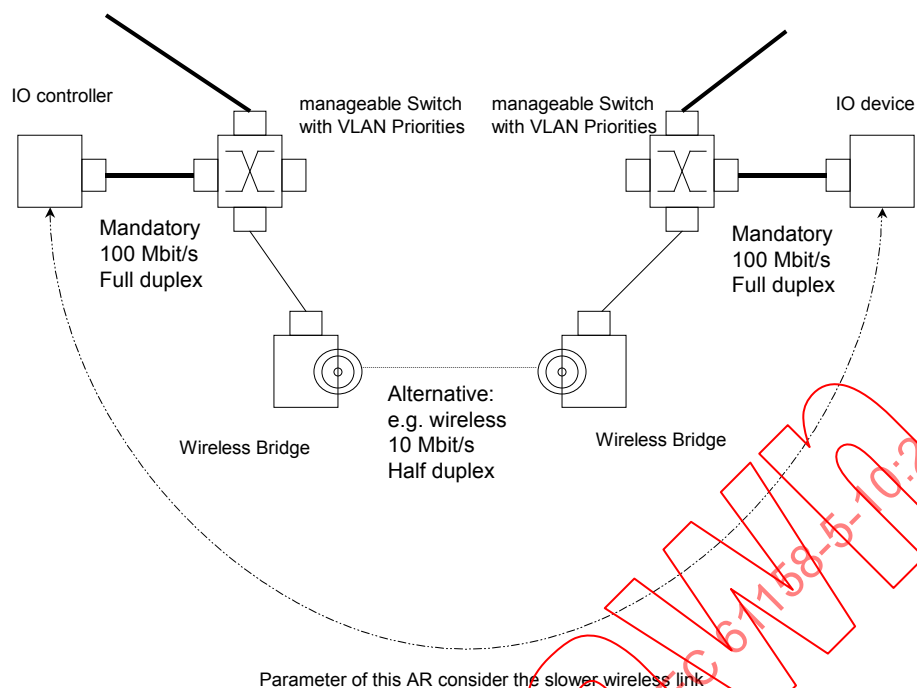


Figure 59 – Example of network topology including slower wireless segments

Figure 60 shows an example closing a line with wireless modules for media redundancy. The shown topology is applicable if the timing required by the application process can be fulfilled.

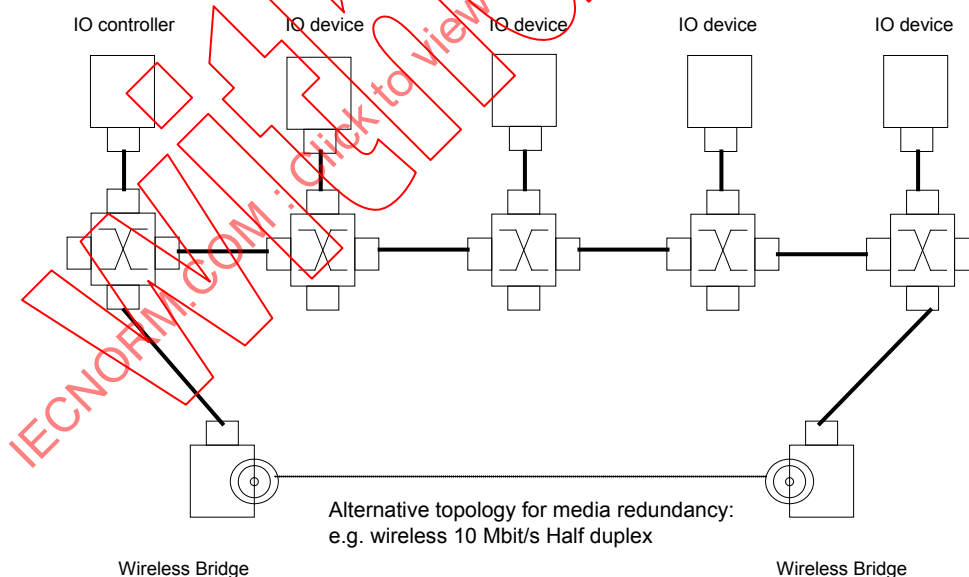


Figure 60 – Example of media redundancy including wireless segments

7.7 Summary of FAL services

7.7.1 IO Device

Table 333 contains the FAL services of the IO device.

Table 333 – FAL services of the IO device

Services	req	ind	rsp	cnf
Alarm	x	x	x	x
Application Ready	x	—	—	x
Connect	—	x	x	—
Prm End	—	x	x	—
Prm Begin	—	x	x	—
Read	—	x	x	—
Write	—	x	x	—
Local Add Diagnosis Entry	x	—	—	x
Local AR Abort	x	x	—	x
Local AR In Data	—	x	—	—
Local Create LogBook Entry	—	x	—	—
Local Data State Changed	—	x	—	—
Local Diagnosis Event	—	x	—	—
Local Get Input IOCS	x	—	—	x
Local Get Output	x	—	—	x
Local Get Time	x	—	—	x
Local New Output	—	x	—	—
Local Remove Diagnosis Entry	x	—	—	x
Local Set AR State	x	—	—	x
Local Set Command	—	x	x	—
Local Set Input	x	—	—	x
Local Set Output IOCS	x	—	—	x
Local Set Provider State	x	—	—	x
Local Set Redundancy	x	—	—	x
Local Set State	x	—	—	x
Local Set Time	x	—	—	x
Local Sync State Info	—	x	—	—
Local SYNCH Event	—	x	—	—
Local Time State Info	—	x	—	—

7.7.2 IO Controller

Table 334 contains the FAL services of the IO controller.

Table 334 – FAL services of the IO controller

Services	req	ind	rsp	cnf
Alarm	x	x	x	x
Application Ready	—	x	x	—
Connect	x	—	—	x
Prm End	x	—	—	x
Prm Begin	x	—	—	x
Read	x	x	x	x
Write	x	x	x	x

Services	req	ind	rsp	cnf
Local AR Abort	x	x	—	x
Local AR In Data	—	x	—	—
Local Create LogBook Entry	—	x	—	—
Local Data State Changed	—	x	—	—
Local Diagnosis Event	—	x	—	—
Local Get Input	x	—	—	x
Local Get Output IOCS	x	—	—	x
Local Get Time	x	—	—	x
Local New Input	—	x	—	—
Local Set AR State	x	—	—	x
Local Set Command	—	x	x	—
Local Set Input IOCS	x	—	—	x
Local Set Output	x	—	—	x
Local Set Provider State	x	—	—	x
Local Set Redundancy	x	—	—	x
Local Set State	x	—	—	x
Local Set Time	x	—	—	x
Local Sync State Info	—	x	—	—
Local SYNCH Event	—	x	—	—
Local Time State Info	—	x	—	—

7.7.3 IO Supervisor

See 7.7.2.

Annex A (informative)

Device instances

The design and construction of physical automation devices is beyond the scope of this specification and therefore vendor specific. However, it provides a model for physical devices to host one or more IO controller or IO devices. These particular IO controller or IO devices are referred to as instances. The physical device is characterized by its device address which is a unique IP specification address assigned to a network interface. The further addressing is provided by means of the underlying connectionless protocol according to Publication C706 of the Open Group. These means are:

- Interface UUID
- Object UUID

The model is provided in Figure A.1. It covers the IO controller and IO device. The gray boxes in Figure A.1 are beyond the scope of this service definition.

NOTE 1 The term "Interface" is used according to Publication C706 of the Open Group. An interface represents a collection of functions or services that can be called by a remote user.

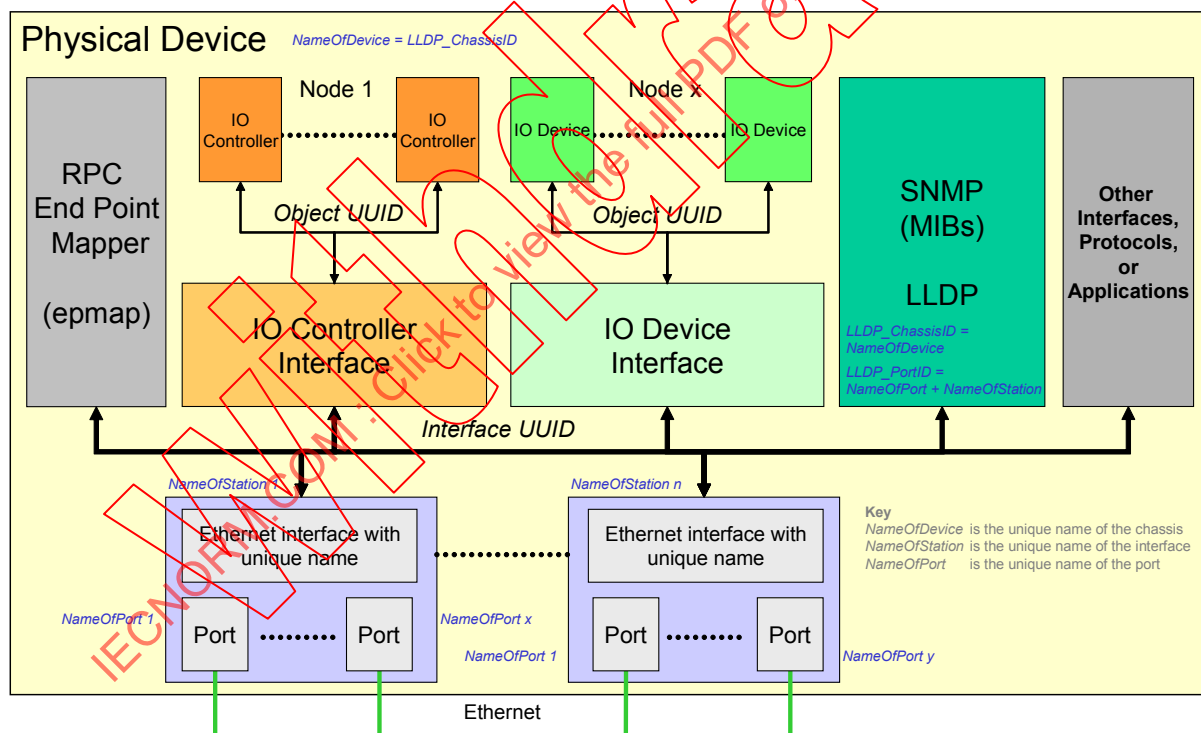


Figure A.1 – Instance model

The Interface UUID provides a means to address an IO supervisor, an IO controller, or an IO device without further configuration. The following Interface UUIDs are defined in 5.3.1.6.1:

- IO Device Interface UUID
- IO Controller Interface UUID
- IO Supervisor Interface UUID
- IO Parameter Server Interface UUID

Furthermore, the epmap interface UUID shall be supported by all IO device classes.

The Object UUID provides a means to address the instance of the IO supervisor, IO controller, IO parameter server, or IO device.

The connectionless DCE RPC protocol according to Publication C706 of the Open Group defines a local instance referred to as End Point Mapper (epmap). The End Point Mapper receives any incoming RPC request at TCP/UDP and transfers the request to the matching interface and object. Therefore, each server has to register its UUIDs before. This information includes:

- Interface with its UUID, Major Version, Minor Version
- Object UUID
- Binding Information such as protocol (UDP/TCP), server IP address, port, 64 octets annotation

NOTE The GSD contains the information about the maximum number of instances a device supports.

Annex B (informative)

Components of an Ethernet interface

The design and construction of physical devices is beyond the scope of this specification and therefore vendor specific. However, an Ethernet interface needs a sum of components:

- RJ45 female connector
- Magnetics
- PHY
- MAC
- μ Controller
- RAM
- ROM

A scheme of components is provided in Figure B.1 and Figure B.2. The gray boxes in Figure B.1 are beyond the scope of this service definition.

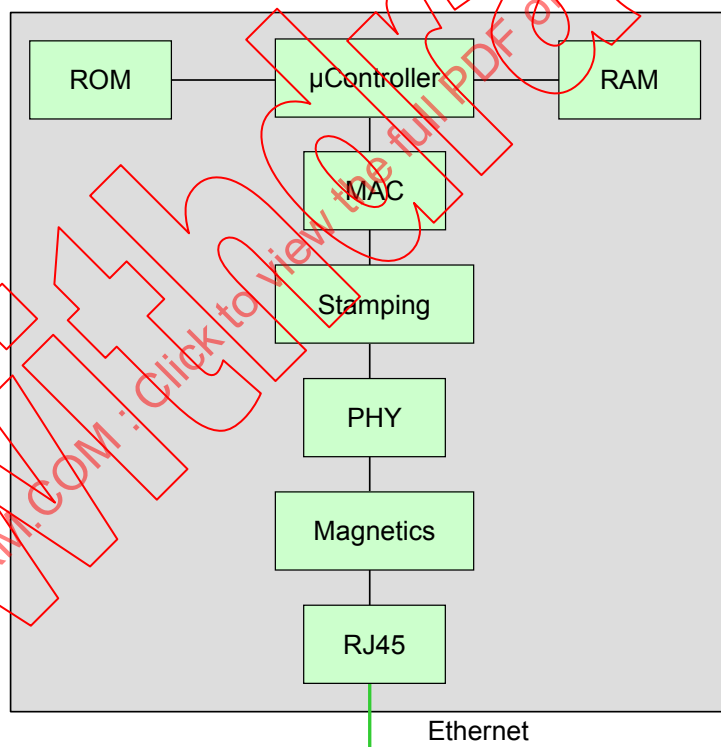


Figure B.1 – Scheme of an Ethernet interface

With the ability for bridging, the structure of the Ethernet interface changes. Figure B.2 shows the new scheme.

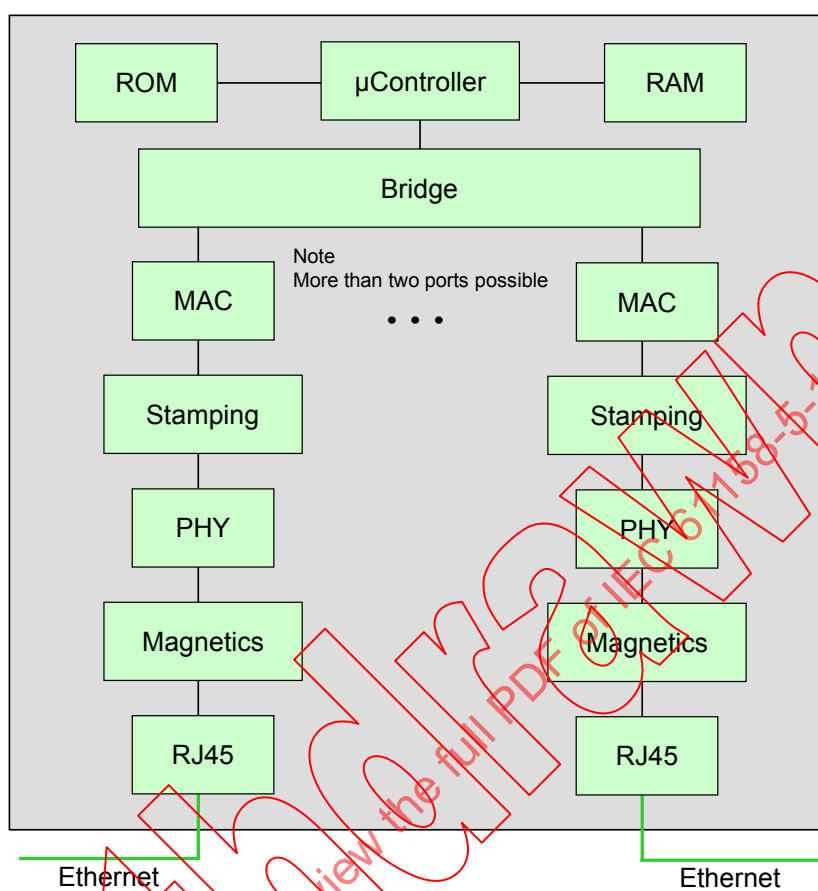


Figure B.2 – Scheme of an Ethernet interface with bridging ability

Figure B.3 shows the scheme using optical media.

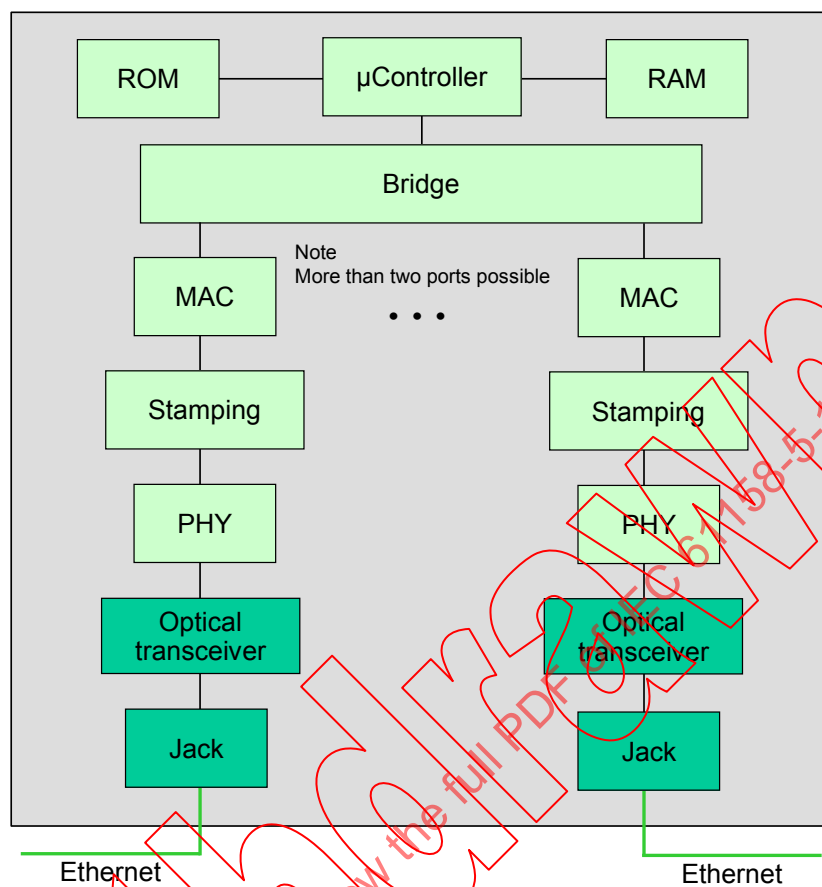


Figure B.3 – Scheme of an Ethernet interface with optical ports

Figure B.4 shows the scheme using radio communication.

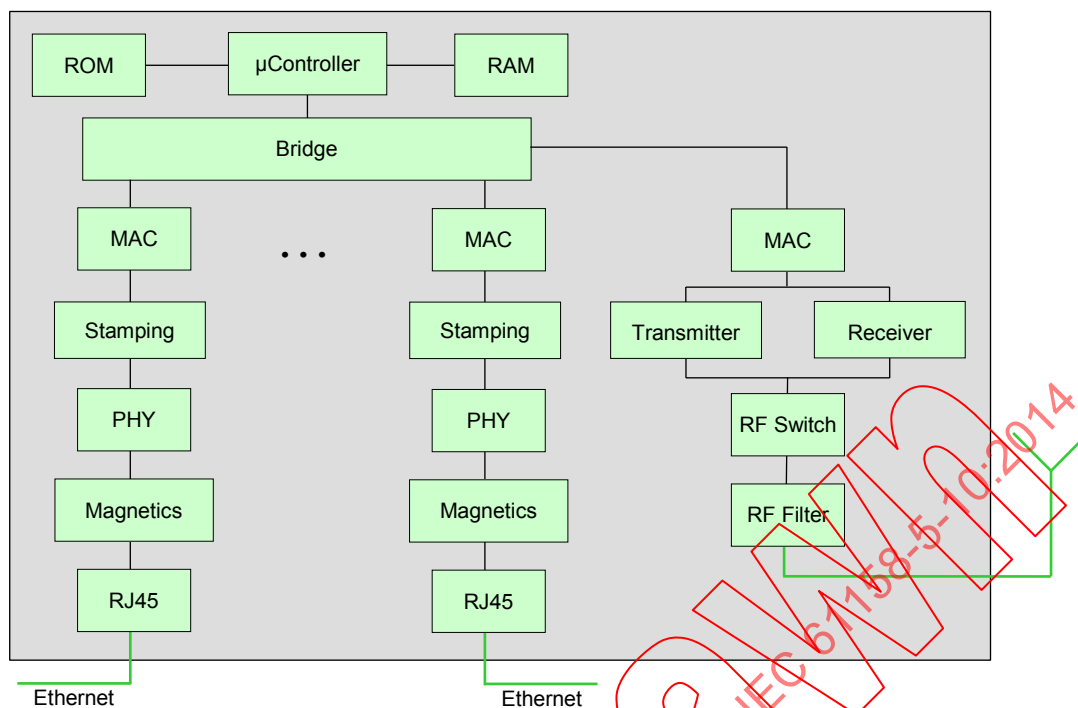


Figure B.4 – Scheme of an Ethernet interface with bridging ability using radio communication

Figure B.5 shows the scheme using radio communication.

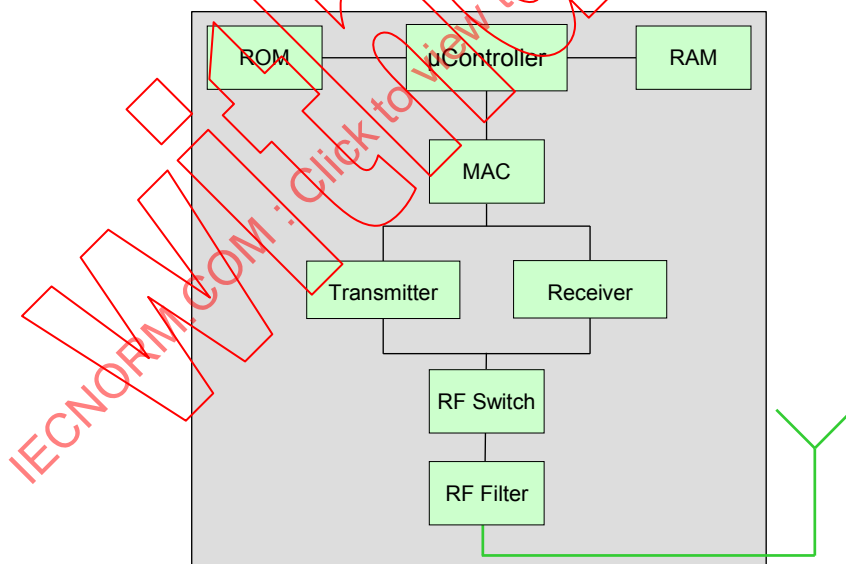


Figure B.5 – Scheme of an Ethernet interface with radio communication

Annex C (informative)

Scheme of MAC address assignment

Figure C.1 shows the scheme of MAC address assignment according to IEEE 802.1D

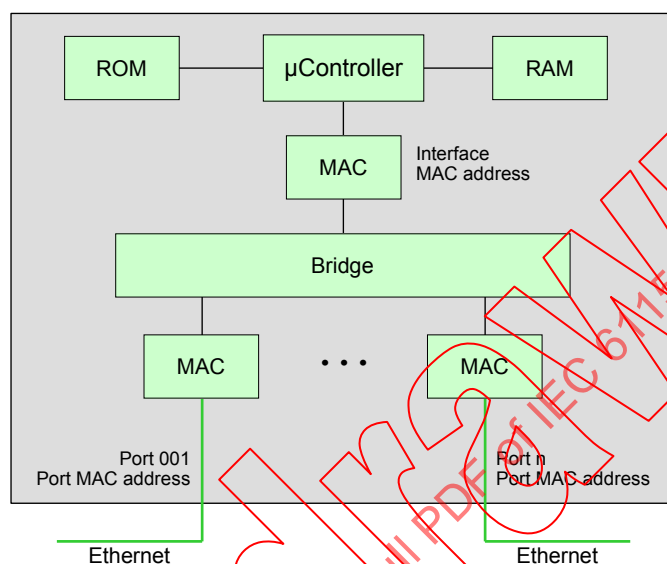


Figure C.1 – Scheme of MAC address assignment

Annex D (informative)

Collection of objects

The objects of an IO device can be read using the read record service and the predefined filters addressed by the record indices. The IO device builds the intersection and search for available objects.

The size of the intersection depends on the AR type, the record index and the IO device. For the implicit AR and the IOS AR for DeviceAccess the intersection is only depending on the record index and the IO device. For an IO AR it also depends on the selected submodules of the AR.

Example Figure D.1 shows the intersection of objects for a read record service with a slot specific index from an IO controller using an IO AR. In this case the IO device should use the "green" intersection to search for e.g. "Diagnosis in channel coding" and send the available data in the read record response.

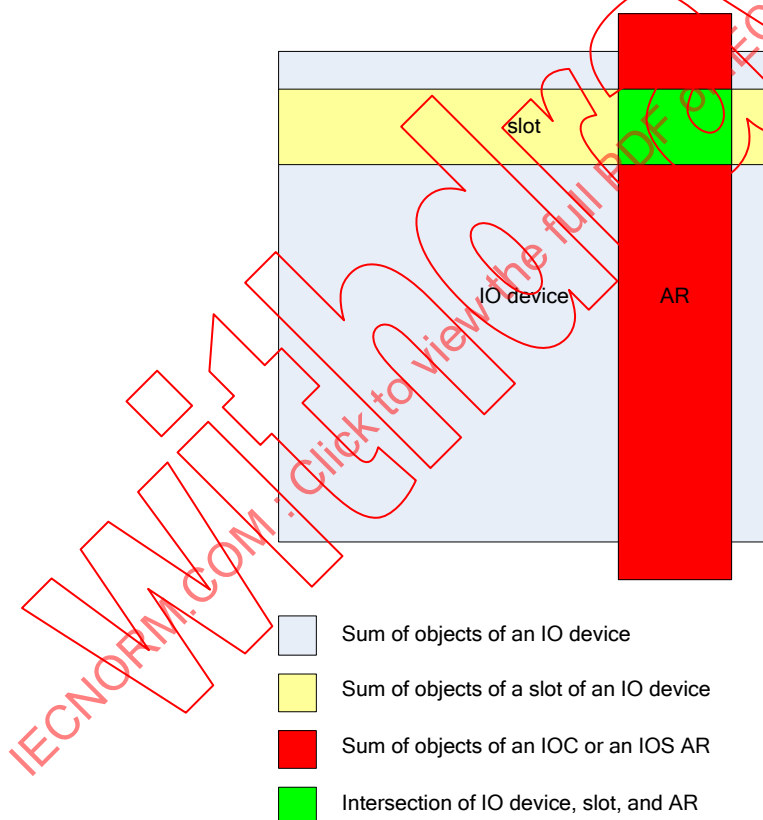


Figure D.1 – Example for an intersection of IO device, slot, and AR

Annex E (informative)

Measurement of the fast startup time

For the comparison of devices the definition of the measurement of the fast startup time is necessary. Figure E.1 shows the reference model for this measurement.

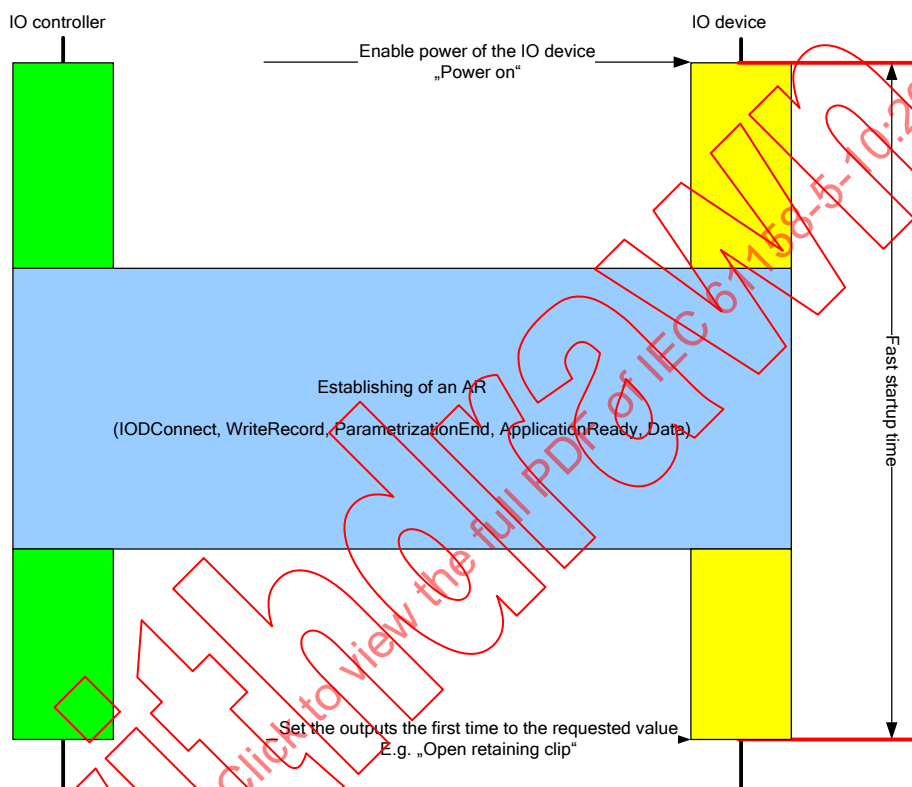


Figure E.1 – Measurement of the fast startup time

Annex F (informative)

Dynamic Frame Packing

An RT Frame is structured according to Figure F.1. The frame itself is encapsulated for Ethernet according to IETF RFC 894 by using the elements destination address, source address, ether type and CRC.

If priority tagging according to IEEE 802.1Q is used, the ether type is set to 0x8100, a priority / VLAN field is following, followed by a second ether type set to 0x8892 indicating an RT frame.

The content of the RT frame are the elements Frame ID, C_SDU and APDU Status.

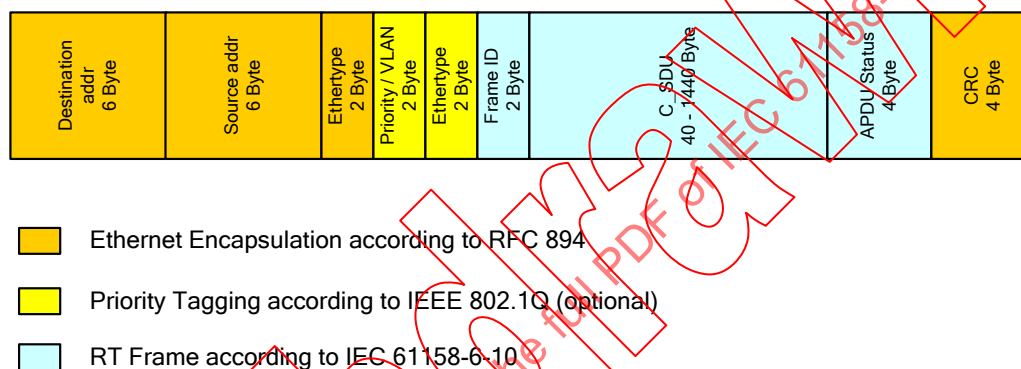


Figure F.1 – Frame Layout

The Frame ID is used to indicate the frame itself, the C_SDU to transport the content of IOCRs (i.e. IO Data, IOXS) and the APDU Status indicates the status of the frame.

The C_SDU may be structured such that it either carries the content of exactly one IOCR or that it carries the content of multiple IOCRs. In the latter case a sub-part of the C_SDU (called sub frame) carries the content of a particular IOCR; the frame is then sub-structured into sub frames.

The reason for using sub-structured frames is minimizing bandwidth and in consequence optimizing performance. As Figure F.1 shows, the overhead of a frame, i.e. the bytes to be transported besides the C_SDU, amounts to 28 bytes; however since InterFrameGap (12 bytes), Preamble (7 bytes) and Sync (1 byte) have additionally to be taken into account, the overhead of a frame sums up to 48 bytes (or 42 bytes if the preamble is shortened). If the C_SDU is in addition smaller than 40 bytes, the delta has to be added additionally.

Thus one can easily see that using sub-structured frames will reduce the bandwidth needed by combining sub-frames, since as shown in Figure F.2 the overhead for a sub frame amounts to 6 bytes. On combining multiple sub frames in one frame the overhead for the frame is only paid once.

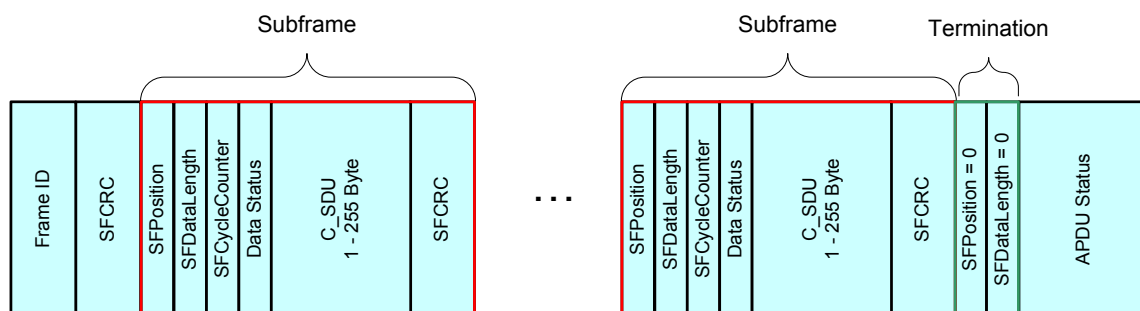


Figure F.2 – Sub frame Layout

Each of the sub frames has a position number (SFPosition), a data length describing the length of the C_SDU (SFDataLength), a cycle counter (SFCycleCounter), a data status (DataStatus), the frame's data itself (C_SDU) and a CRC for the sub frame (SFCRC).

The position number starts with 1 and increases with every sub frame by 1 (the second sub frame thus carries the position number 2 and so on). The list of sub frames is terminated by a special sub frame having the position number 0.

The data status of a sub frame indicates its data status. The data status within the frame's APDU status indicates the frame's data status. If the frame consists of sub frames, the frame's data status can be safely ignored. It is feasible to set the frame's data status to a static value (e.g. DataValid = valid, ProviderState = Run, ProblemIndicator = Normal Operation, SubframeSenderMode = 0x00, Ignore = 0x01 ...).

Each of the sub frames is from an applications point of view comparable to a complete frame, since it has a local APDU Status (i.e. cycle counter and data status) and a local C_SDU (for IO data and IOXS). It is however restricted in the maximum size of the C_SDU; the maximum is 255 bytes (compare to a complete frame; there the maximum is 1 440 bytes).

Sub-structured frames are built in two ways:

Dynamic Frame Packing

The sub-structured frame will be built by a chain of adjacent IO devices.

Up to 63 sub frames are supported in a dynamically packed frame.

End to End

The sub-structured frame will be packed completely by one IO device. A possible use case is for example a Multi-IO device (e.g. a device hosting multiple IO device instances), where the IOCRs to the multiple instances are packed together by this IO device. Another use case is for example a device used for integration of another fieldbus (e.g. gateway). Here the IOCRs to the subordinated fieldbus devices are packed together by having one sub frame per subordinated IO device.

Up to 127 sub frames are supported in such an end to end frame.

End to End may be used for input and output data as shown in Figure F.3.

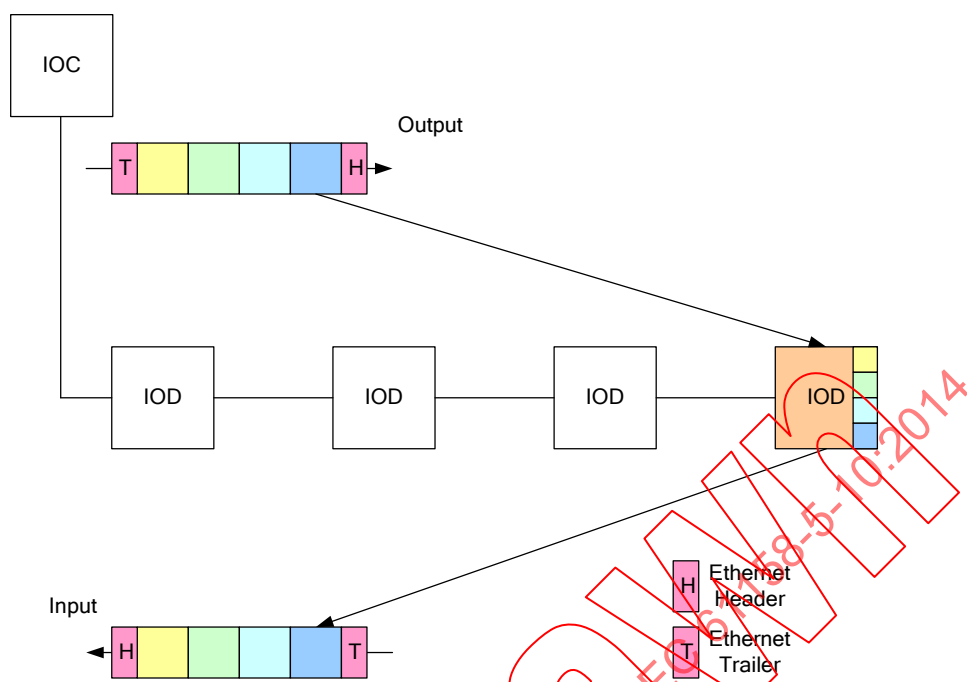


Figure F.3 – End to End

The dynamic frame packing may be used for input and output data as shown in Figure F.4.

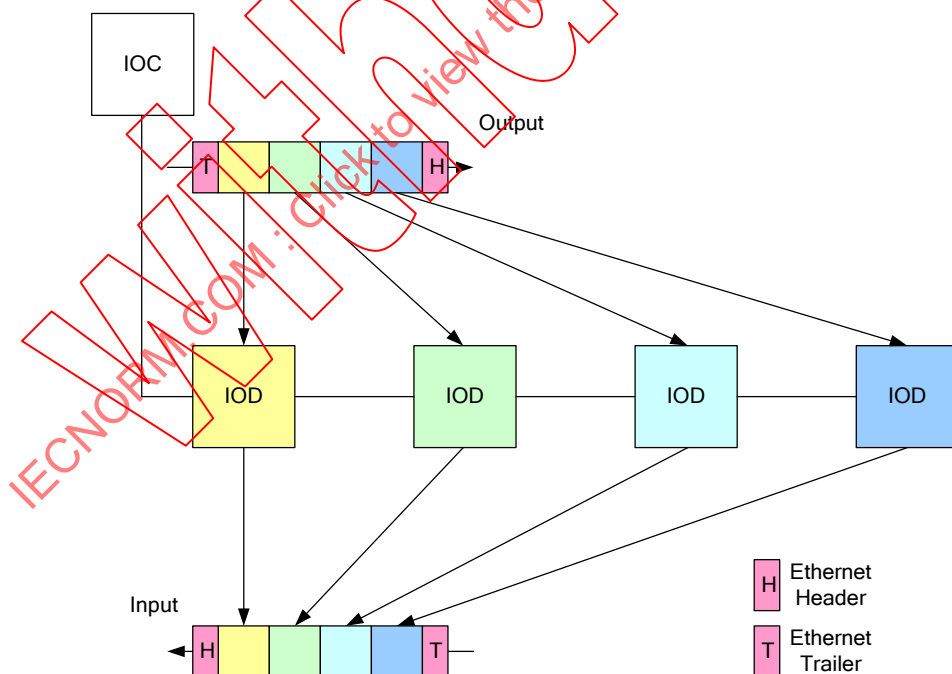


Figure F.4 – Dynamic frame packing

As an additional optimization the output frame (Provider from the IO controller point of view) shall be truncated as shown in Figure F.5.

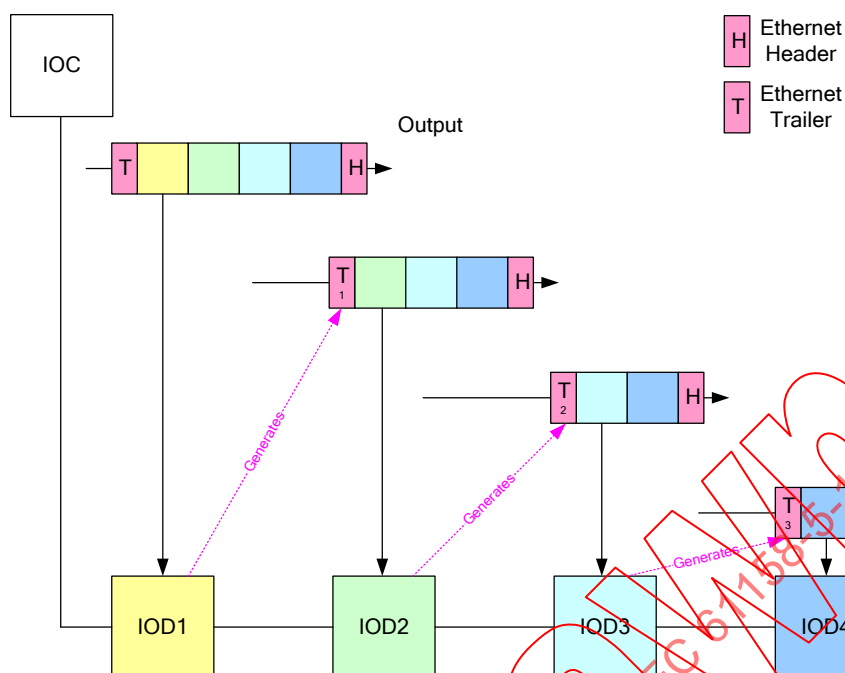


Figure F.5 – Dynamic frame packing – Truncation of outputs

Every IO device is triggered “downstream” (as seen by the IO controller) through the frame as indicated by the frame ID or additionally by the MC fast forwarding address. While receiving the frame on one port, the IO device propagates the frame downstream on the appropriate port. The frame propagated downstream is shortened by cutting the frame at the sub frame position dedicated to this IO device and appending a new Ethernet trailer with an updated CRC computed for the shortened frame. The sub frame dedicated to this IO device is received locally.

Figure F.6 shows outbound packing from its timing behavior.

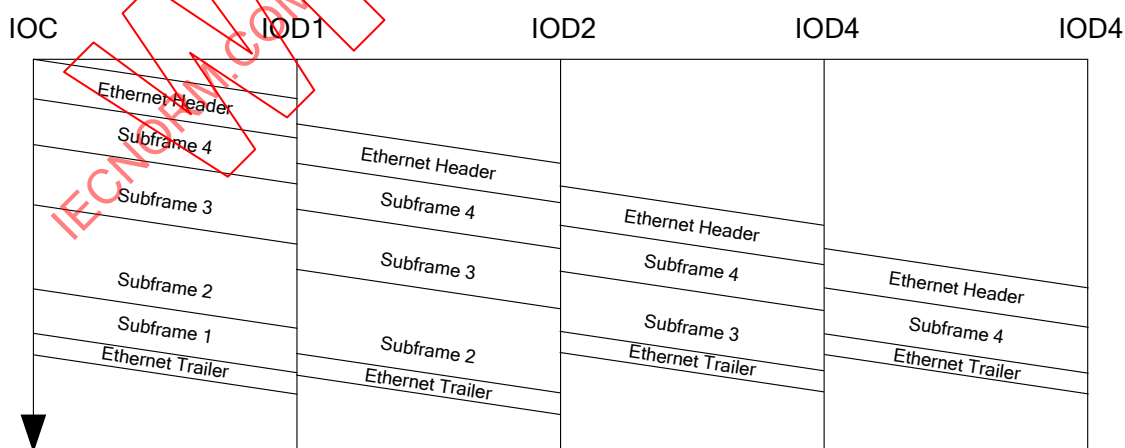


Figure F.6 – Dynamic frame packing – Outbound Pack

It is easily to see that the particular sub frames are received by the IO devices downstream at almost the same time.

The input frame (Consumer from the IO controller point of view) shall be concatenated as shown in Figure F.7 for optimization.

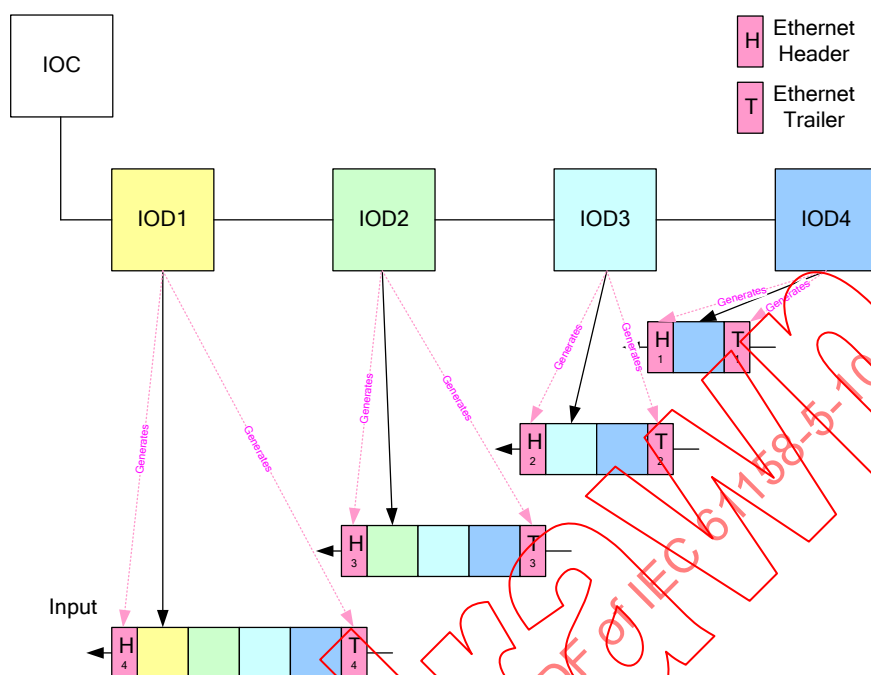


Figure F.7 – Dynamic frame packing – Concatenation of inputs

Every IO device has a locally defined start time to issue the input frame. While sending the frame on the appropriate upstream port, the IO device awaits receiving the frame of the IO device downstream on the other port (indicated by Frame ID or additionally by the MC fast forwarding address). The header and trailer of the received frame are dropped and the received sub frame(s) are appended at the appropriate sub frame position. Then an Ethernet trailer with CRC computed for the complete frame content is appended.

Since all IO devices start sending more or less at the same time, IOD3 appends for example the sub frame of IOD4, IOD2 the sub frames of IOD3 and IOD4 (having been concatenated by IOD3), IOD1 the sub frames of IOD2, IOD3 and IOD4 (having been concatenated by IOD2).

Since the Ethernet header of an IO device (e.g. H1 of IOD4) is removed in the next IO device upstream (e.g. IOD3) the time needed to send the header bytes from IOD4 to IOD3 can be optimized. It is sufficient, when the first byte of the sub frame data is available in the IO device, after the H2 of IOD3 and the sub-frame data related to IOD3 has been sent already. In this case the sub frame data of IOD4 can be docked “on the fly” to the frame currently leaving IOD3. This applies similar to IOD1 and IOD2.

Figure F.8 shows inbound packing from its timing behavior.

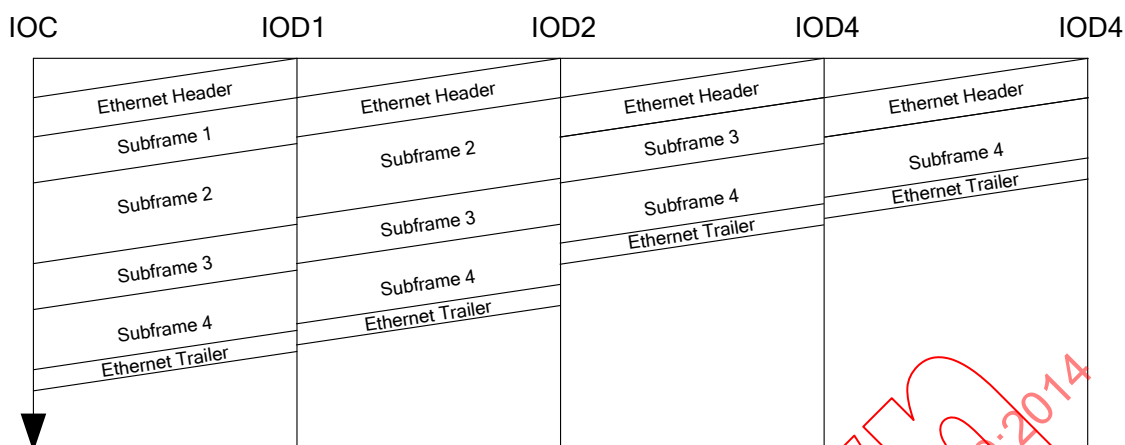


Figure F.8 – Dynamic frame packing – Inbound Pack

The Frame Send Offset of the particular frames is computable by using the following means: The first byte of the header of some IO device (e.g. IOD4) will appear after LineDelay + local Bridge Delay in the IO device upstream (e.g. IOD3). Thus time needed to send the own sub frame (e.g. the sub frame of IOD3) has to compensate LineDelay + local BridgeDelay + Frame Send Offset.

This leads to the following formula:

$$FrameSendOffset_{right} + Length_{subframe} * 80 ns \geq FrameSendOffset_{left} + LineDelay_{to right} + BridgeDelay_{left} \quad (F.1)$$

where

$FrameSendOffset_{right}$ is the frame send offset right
 $Length_{subframe}$ is the length of sub frame
 $FrameSendOffset_{left}$ is the frame send offset left
 $LineDelay_{to right}$ is the line delay to right
 $BridgeDelay_{left}$ is the bridge delay left

NOTE 1 Be aware, that if the Preamble is adjusted differently through the DFP group, the formulas have to be adjusted.

If one wants to have the same Frame Send Offset (e.g. 0) through the DFP group, the formula can be transformed to

$$Length_{subframe} \geq \frac{LineDelay_{to right} + BridgeDelay_{left}}{80 ns} \quad (F.2)$$

where

$Length_{subframe}$ is the length of sub frame
 $LineDelay_{to right}$ is the line delay to right
 $BridgeDelay_{left}$ is the bridge delay left

NOTE 2 As a hypothetical sample take for LineDelay 500 ns and for BridgeDelay 600 ns. Then the length of the sub frame has to be greater or equal to 14 bytes. Note that the sub frame definition allows GAP (i.e. empty space at the end of the sub frame). Thus IO devices with less IO data can be compensated by adding a GAP.

Every IO device has a locally defined start time to issue the input frame. While sending the frame on the appropriate upstream port, the IO device awaits receiving the frame of the IO

device downstream on the other port (indicated by Frame ID in the payload or in the MC fast forwarding address). The header and trailer of the received frame are dropped and the received subframe(s) are appended at the appropriate subframe position. Then an Ethernet trailer with CRC computed for the complete frame content is appended.

If the frame from the downstream neighbor doesn't arrive in time (i.e. is too late to be appended), a DFP late error is raised and the latest received frame (from an earlier cycle) is appended instead. If no frame has been received up to that point, a surrogate frame is appended.

The DFP late error allows detecting wrong engineering timings for the DFP group. To reduce the alarm load a DFP late error will be issued by a particular IO device only once per second.

The surrogate frame keeps the structure of the frame valid, even if the source of a particular sub-frame is vanished. A surrogate frame has thus correct values for position number, data length, cycle counter, data status and CRC (the outer shell of the sub-frame is thus kept intact) and have a null'd C_SDU (i.e. all IOXS are set to Bad). A surrogate frame is treated in the CPM as "sub-frame not received" but updates the data, thus the WDT/DHT on the IO controller expires also. In consequence the application sees IOXS = "Bad" immediately and after WDT/DHT an application relation breakdown.

To detect a station loss within a DFP group there are two possible mechanisms depending on the parameters:

- If the IO controller works on subframe level the station loss is detected by older/newer comparison of the subframe cycle counter
- If the IO controller does not support subframe level, the older/newer detection is delegated to the IO devices of the pack group. This is called distributed watchdog. If a particular IO device detects the loss of a subframe through older/newer detection using the value of `SFIOCRProperties.DistributedWatchDogFactor` (see 7.3.8.2.2), a surrogate frame is appended. To allow the IO controller detecting the station loss safely (since done by software), the surrogate frame is appended at least for `SFIOCRProperties.RestartFactorForDistributedWD` cycles (see 7.3.8.2.2).

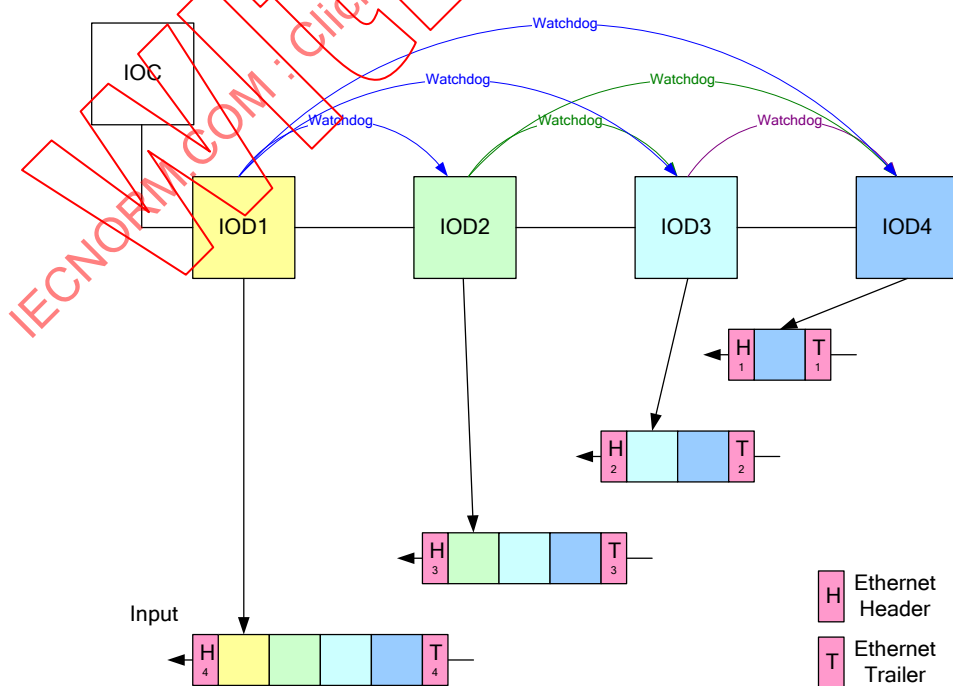


Figure F.9 – Dynamic frame packing – Distributed watchdog

The interrelation between the IO CR for each IO device and the dynamically packed frame is expressed in Figure F.10.

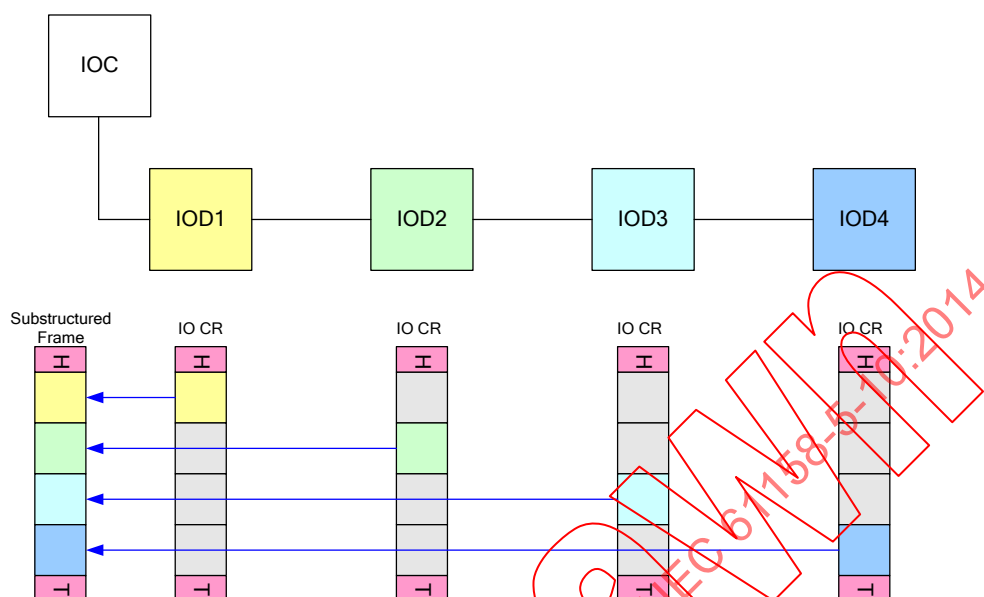


Figure F.10 – Interrelation between IO CR and dynamically packed frame

The parameterization of the IO CR for each IO device describes the structure of the substructured frame, i.e. the number of sub frames, their position and their length (i.e. the grayed boxes in each IO CR). For a particular IO device only the content of its own sub frame is described through the CRL's attributes Related IO Data objects and Related IOCS. It has no knowledge over the content of other sub frames; only the layout of the complete frame is known. Only the IO controller has the knowledge over both content and layout of the complete sub-structured frame.

Annex G (informative)

Building IR Data

This annex explains how the IR data is built by an engineering system.

As stated in 6.3.9.2 the IR Data consists of the three major elements PD IR Global Data, PD IR Frame Data (optional) and PD IR Begin End Data.

The PD IR Global Data consists of an IR Data UUID. The IR Data UUID is to be regenerated by the engineering system, if any of the parameter changes. The other attributes of the IR Global Data describe the delays of both bridge and ports. That information is built solely out of GSDML elements, i.e. the GSD of the IO device states the appropriate delays.

The list of PD IR frame data elements has to be built for IO devices which support RT_CLASS_3 with frame data elements only; this is again stated by the GSD.

The list contains for all frames out of the frame ID range 3 (see Table 276), which the device shall feed in or forward, path information (i.e. on which ports a particular frame is to be forwarded) and time information (i.e. when a particular frame is to be sent). The path information is described by the attribute Tx Port Group, the time information by Frame Send Offset.

Tx Port Group describes, on which ports a particular frame is to be scheduled.

The PD IR Begin End Data contains begin and end of the particular RED phases build.

For the further understanding the propagation delay is essential and thus defined by the following formulae:

$$\begin{aligned}
 \text{PropagationDelay} &= \text{MaxBridgeDelay} + \text{LineDelay} \\
 \text{LineDelay} &= \text{MaxPortTxDelay}_{\text{Sender}} + \text{CableDelay} + \text{MaxPortRxDelay}_{\text{Receiver}} \\
 \text{CableDelay} &= \text{CableLength} * \frac{5,48 \text{ ns}}{m}
 \end{aligned}
 \tag{G.1}$$

where

<i>PropagationDelay</i>	is the propagation delay
<i>MaxBridgeDelay</i>	is the max bridge delay
<i>LineDelay</i>	is the line delay
<i>MaxPortTxDelay</i> _{Sender}	is the max port Tx delay
<i>CableDelay</i>	is the cable delay
<i>MaxPortRxDelay</i> _{Receiver}	is the max port Rx delay
<i>CableLength</i>	is the cable length

The particular delays are explained according to Figure G.1.

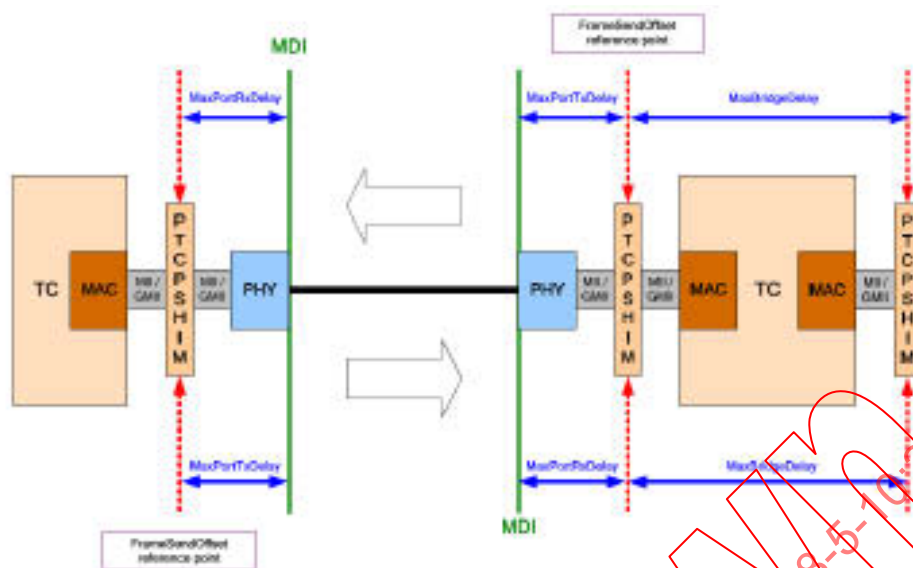


Figure G.1 – Bridge- and LineDelay

For the further explanation of building the path and time information for all frames, let assume the following topology used as sample as stated in Figure G.2.

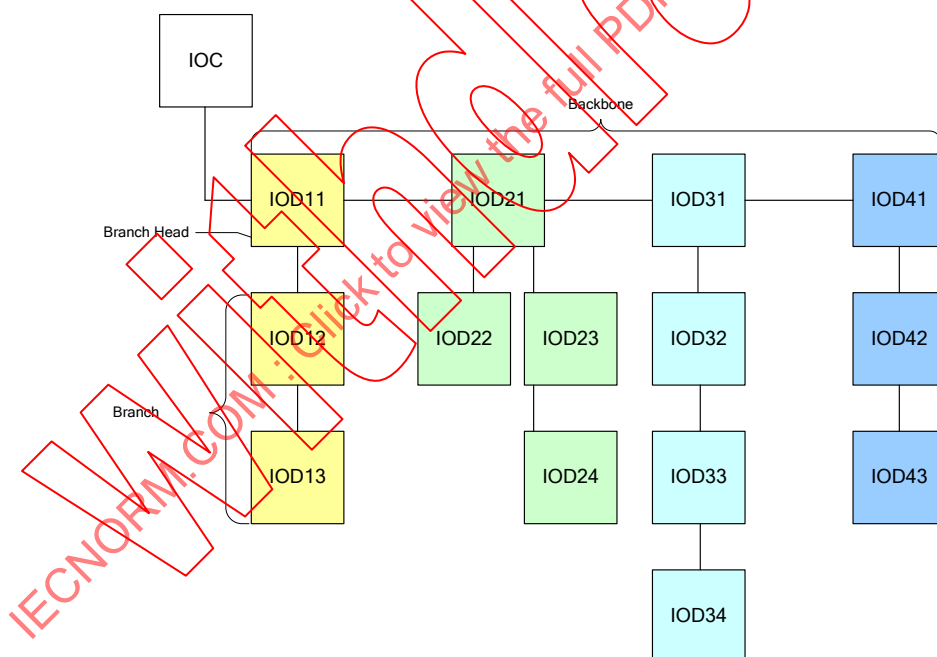


Figure G.2 – Sample Topology

The topology show is a so called “comb topology”. It shows a linear “backbone” (the linear topology IOD11+IOD21+IOD31+IOD41) and several linear branches (such as IOD12+IOD13, IOD22, IOD23+IOD24 or IOD42+IOD43+IOD44). Note that every network is reducible onto such a topology; a linear topology only has a backbone, a star topology has only branches.

Within a particular branch the planning of the frames should honor the slipstream effect. If the frames for a particular linear topology (e.g. a branch) are send by the IO controller in reversed order (frame to the last IO device in line is send first, frame to the first IO device is send last), the frames will reach the IO devices in about the same time.

The end of the transmission is running to the left side, if the propagation delay is smaller than the average size of the frames send. In other words, as shown by Figure G.3, the frame to the last IO device in line is received earlier than the frame to the first IO device.

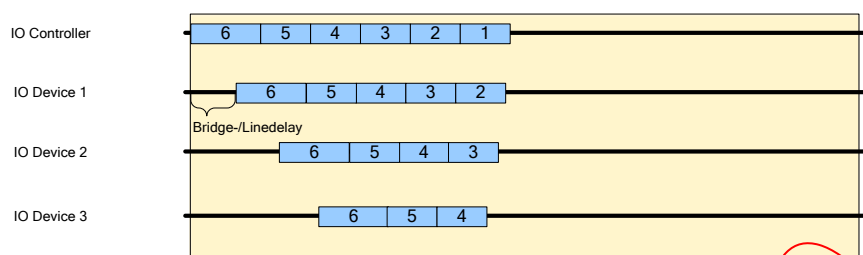


Figure G.3 – Slip Stream Effect downstream

Figure G.3 shows another interesting effect. Since the first frame in an IO device is received with an offset of the sum of all propagation delays, that bandwidth is usable for the GREEN traffic class. This is the reason for the IR Begin End Data: determining begin and end of the RED phase and thus getting more GREEN bandwidth. The farther downstream, the smaller is RED phase. The begin of the RED phase is determined by the following formula:

$$RED_{Begin} = BranchOffset + \sum_{IODupstream} PropagationDelay \quad (G.2)$$

where

RED_{Begin} is the begin of RED phase
 $BranchOffset$ is the branch offset
 $PropagationDelay$ is the propagation delay
 $IODupstream$ is the IO dupstream

The BranchOffset is determined by the following assumption. The IO controller has to send out all frames on the backbone. From this frame stream a particular section is determined for a particular branch. Again the slip stream effect is used in the broader sense (see Figure G.4).

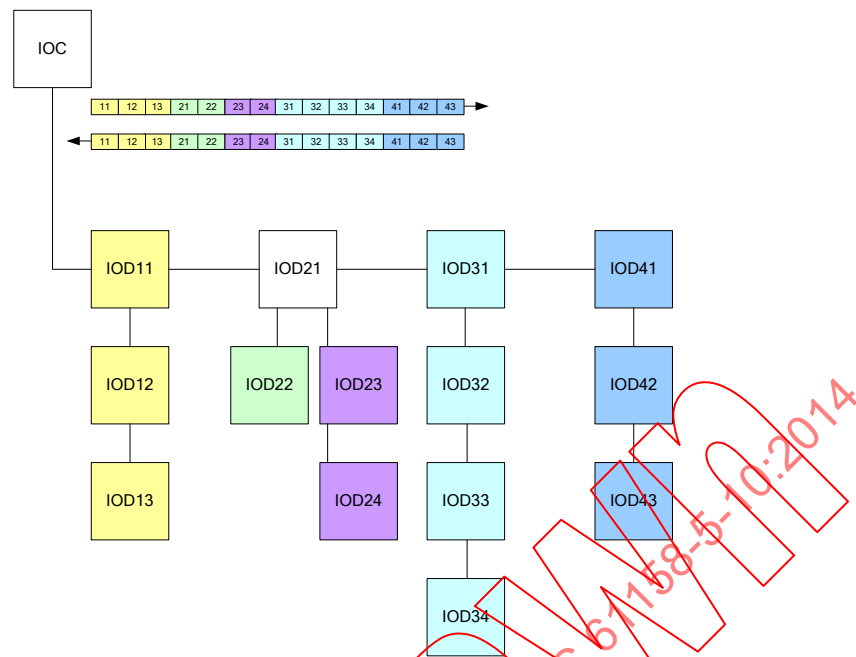


Figure G.4 – Using the slip stream effect in a comb topology downstream

For a device with more branches it is regardless, which branch is sent earlier or later (e.g. IOD22 or IOD23+IOD24).

The BranchOffset is thus determined by the time, when the first frame for the branch reaches the device where the branch branches off; e.g. when the frame for IOD31 reaches IOD32.

The end of the RED phase is determined by the end of transmission. The frames should be scheduled such, that a stream of data is flowing; i.e. the frames are transmitted as packed as possible with just the Inter Frame GAP (12 bytes), the preamble (7 bytes or potentially shortened to 1 byte) and the start frame delimiter (1 byte).

Thus the end of the RED phase is determined at a particular port as adding up the size of all frames to be transmitted or else received, and adding the space in between. The resulting number of bytes multiplied with 80 ns is the time to add to the RED_{Begin}.

For the frames in upstream direction (i.e. from the IO devices to the IO controller) the slip stream effect is used in the other direction. Here all IO devices within a branch may start sending their frame at offset 0. If the propagation delay between 2 IO devices is smaller than the minimum frame size, the frames reaching the branches head will have no gaps (besides the gap enforced by Ethernet, i.e. InterFrameGap, preamble and start frame delimiter). Thus a stream without gaps will reach the branches head.

Thus begin of the RED phase is set to 0 in all IO devices along a branch. The end of the RED phase is then sent to the end of transmission or receiving (again determined by the stream size multiplied with 80 ns).

Within the branches head, the stream and the frame of the head itself are to be scheduled such, that a gap less stream is formed on the backbone. Thus the stream received from the branch has to wait for its injection into the backbone stream. The nearer a particular branches head to the IO controller is, the earlier he sends its data upstream directly followed by the data of its branch(es). Again for a branch head with multiple branches the order of the branches is regardless.

The particular Frame Send Offset within the PD IR Frame Data Elements is determined by:

- Upstream:
 - set to 0 in branches
 - set to the particular RED_{begin} at this port and adding the number of bytes in the stream before this particular frame (with respect to the gaps between the frames) multiplied with 80 ns
- Downstream:
 - set to the particular RED_{begin} at this port and adding the number of bytes in the stream before this particular frame (with respect to the gaps between the frames) multiplied with 80 ns.

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-10:2014

Bibliography

IEC 60870 (all parts), *Telecontrol Equipment and Systems*

IEC 61131-1, *Programmable controllers – Part 1: General information*

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

IEC 61158-2:2014, *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61375-1, *Electronic railway equipment – Train communication network (TCN) – Part 1: General architecture*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

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

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

ISO/IEC 8877, *Information technology – Telecommunications and information exchange between systems – Interface connector and contact assignments for ISDN Basic Access Interface located at reference points S and T*

ISO 15745-4, *Industrial automation systems and integration – Open systems application integration framework – Part 4: Reference description for Ethernet-based control systems Amendment 1: PROFINET profiles*

IECNORM.COM : Click to view the full PDF of IEC 61158-5-10:2014

SOMMAIRE

AVANT-PROPOS	627
INTRODUCTION.....	629
1 Domaine d'application	630
1.1 Généralités.....	630
1.2 Spécifications.....	631
1.3 Conformité	631
2 Références normatives.....	631
3 Termes, définitions, abréviations, symboles et conventions	634
3.1 Termes et définitions référencés	634
3.2 Termes et définitions complémentaires pour périphérie décentralisée.....	635
3.3 Termes et définitions complémentaires pour la redondance des supports.....	643
3.4 Abréviations et symboles.....	644
3.5 Conventions	647
4 Concepts.....	654
5 ASE Data type.....	654
5.1 Généralités.....	654
5.2 Définition formelle des objets de types de données.....	660
5.3 Types de données définis pour la FAL.....	662
5.4 Spécification de service de l'ASE Data type.....	685
6 Modèle de communication pour services communs.....	685
6.1 Concepts.....	685
6.2 Types de données ASE.....	686
6.3 Eléments de service application.....	687
7 Modèle de communication pour périphérie décentralisée.....	806
7.1 Concepts.....	806
7.2 Types de données des ASE.....	830
7.3 ASE.....	830
7.4 Comportement d'un appareil E/S	1144
7.5 Comportement d'un appareil de commande E/S	1207
7.6 Caractéristiques d'application.....	1212
7.7 Résumé des services de FAL	1214
Annexe A (informative) Instances d'appareil	1217
Annexe B (informative) Composants d'une interface Ethernet	1219
Annexe C (informative) Schéma d'une affectation d'adresse MAC	1224
Annexe D (informative) Ensemble d'objets	1225
Annexe E (informative) Mesure du temps de démarrage rapide	1227
Annexe F (informative) Condensation de trame dynamique.....	1228
Annexe G (informative) Construction des IR Data	1239
Bibliographie.....	1244
Figure 1 – Exemple de hiérarchie de la classe de types de données "Data type"	655
Figure 2 – Relation de date de NetworkTime.....	677
Figure 3 – Architecture de communication des ASE de FAL.....	686
Figure 4 – Applications de PTCP	706
Figure 5 – Mesure de dérive d'horloge	719

Figure 6 – Synchronisation plurielle	720
Figure 7 – Dépendances du diagnostic de redondance de supports	726
Figure 8 – Exemple de périodes en un port local.....	790
Figure 9 – Exemple de communication entre appareils de commande et appareils de terrain.....	807
Figure 10 – Exemple de communication entre une station d'ingénierie et plusieurs appareils de commande et de terrain	808
Figure 11 – Exemple de communication entre appareils de terrain et une station serveur	808
Figure 12 – Exemple de communication entre appareils de terrain.....	809
Figure 13 – Unités structurales d'un API arbitraire d'un appareil E/S (général).....	811
Figure 14 – Exemple 1 d'unités structurales pour interfaces et ports au sein de l'API 0	812
Figure 15 – Exemple 2 d'unités structurales pour interfaces et ports au sein de l'API 0	813
Figure 16 – Hiérarchie d'identification	816
Figure 17 – Vue d'ensemble de processus d'application	819
Figure 18 – Appareil E/S avec des AP, baies et sous-baies	820
Figure 19 – Processus d'application avec objets de processus d'application (APO)	824
Figure 20 – Accès à un APO distant	825
Figure 21 – Accès à un APO distant pour association fournisseur/consommateur	826
Figure 22 – Exemple d'une AR avec deux AREP.....	827
Figure 23 – Architecture de communication des ASE de FAL	830
Figure 24 – Relation d'un objet Record Data à un seul objet réel	831
Figure 25 – Relation d'un objet Record Data à deux objets réels	832
Figure 26 – Vue d'ensemble des interactions de services des ASE IO	855
Figure 27 – Classification de la sévérité du diagnostic et de la maintenance.....	886
Figure 28 – Diagramme de transition d'états DIAG_DIAG	922
Figure 29 – Diagramme de transition d'états DIAG_MR	926
Figure 30 – Diagramme de transition d'états DIAG_MD	930
Figure 31 – Diagramme de transition d'états DIAG_QUALIFIED.....	934
Figure 32 – Exemple de modèle de ressources en une source d'alarmes	955
Figure 33 – Modèle de base pour les applications isochrones	1022
Figure 34 – Modèle général d'application isochrone (exemple: CACF == 1)	1023
Figure 35 – Modèle général d'application isochrone (exemple: CACF == 2)	1024
Figure 36 – Relations d'ASE dans un appareil E/S fonctionnant en mode isochrone pour un sous-module	1032
Figure 37 – Diagramme de transition d'états d'ISOM_SYNC	1034
Figure 38 – Diagramme de transition d'états d'ISOM_OUT.....	1037
Figure 39 – Diagramme de transition d'états d'ISOM_IN	1042
Figure 40 – Affectation d'une relation de communication à une relation entre applications	1112
Figure 41 – Relation entre applications implicite	1116
Figure 42 – Exemple de relation entre applications E/S (IO AR) (un à un).....	1119
Figure 43 – Exemple de relation entre applications E/S (IO AR) (un à plusieurs)	1120
Figure 44 – Vue d'ensemble des diagrammes d'états d'ASE pour un appareil E/S	1145
Figure 45 – Diagramme de transition d'états DEVSM.....	1148

Figure 46 – Diagramme de transition d'états REM_CHK	1155
Figure 47 – Diagramme de transition d'états LOC_LNK	1162
Figure 48 – Diagramme de transition d'états FOMR	1168
Figure 49 – Diagramme de transition d'états FOMD	1171
Figure 50 – Diagramme de transition d'états FODIAG	1173
Figure 51 – Diagramme de transition d'états RSMSM	1178
Figure 52 – Gestion de la propriété	1183
Figure 53 – Diagramme de transition d'états OWNSM	1186
Figure 54 – Diagramme de transition d'états ASSSM	1186
Figure 55 – Diagramme de transition d'états PLUGSM	1199
Figure 56 – Diagramme de transition d'états PULLSM	1202
Figure 57 – Diagramme de transition d'états SYNC_DIAG	1204
Figure 58 – Diagramme d'états CTLSM	1209
Figure 59 – Exemple de topologie de réseau comprenant des segments sans fils plus lents	1213
Figure 60 – Exemple de redondance de supports incluant des segments sans fils	1214
Figure A.1 – Modèle d'instance	1218
Figure B.1 – Schéma d'une interface Ethernet	1220
Figure B.2 – Schéma d'une interface Ethernet avec capacité de pontage	1220
Figure B.3 – Schéma d'une interface Ethernet avec ports optiques	1221
Figure B.4 – Schéma d'une interface Ethernet avec capacité de pontage en cas d'utilisation de communication radio	1222
Figure B.5 – Schéma d'une interface Ethernet avec communication radio	1223
Figure C.1 – Schéma d'affectation d'adresse MAC	1224
Figure D.1 – Exemple pour une intersection d'appareil E/S, baie et AR	1226
Figure E.1 – Mesure du temps de démarrage rapide	1227
Figure F.1 – Présentation de trame	1228
Figure F.2 – Présentation de sous-trame	1229
Figure F.3 – End to End (de bout en bout)	1231
Figure F.4 – Condensation de trame dynamique	1231
Figure F.5 – Condensation de trame dynamique – Troncature des sorties	1232
Figure F.6 – Condensation de trame dynamique – Condensation sortante	1233
Figure F.7 – Condensation de trame dynamique – Concaténation des données d'entrée ..	1234
Figure F.8 – Condensation de trame dynamique – Condensation entrante	1235
Figure F.9 – Condensation de trame dynamique – Chien de garde distribué	1237
Figure F.10 – Interrelation entre IO CR et trame dynamiquement condensée	1238
Figure G.1 – BridgeDelay et LineDelay	1240
Figure G.2 – Topologie échantillon	1241
Figure G.3 – Effet de sillage aval	1241
Figure G.4 – Utilisation de l'effet de sillage dans une topologie comp aval	1242
Tableau 1 – Eléments de description de diagramme d'états	651
Tableau 2 – Description d'éléments de diagramme d'états	651
Tableau 3 – Conventions utilisées dans les diagrammes d'états	651

Tableau 4 – Conventions utilisées pour les services dans les diagrammes d'états	652
Tableau 5 – Vue d'ensemble des types de données	658
Tableau 6 – Octets V2	663
Tableau 7 – Octets L2	663
Tableau 8 – Octets E2	666
Tableau 9 – Plage de valeurs de E2	666
Tableau 10 – Octets Unipolar2.16	667
Tableau 11 – Plage de valeurs de Unipolar2.16	667
Tableau 12 – Plage de valeurs de N2	668
Tableau 13 – Plage de valeurs de N4	669
Tableau 14 – Plage de valeurs de X2	670
Tableau 15 – Plage de valeurs de X4	671
Tableau 16 – Plage de valeurs de C4	672
Tableau 17 – Plage de valeurs de T2	673
Tableau 18 – Plage de valeurs de T2	674
Tableau 19 – Plage de valeurs de D2	675
Tableau 20 – Plage de valeurs de R2	675
Tableau 21 – UUID pour périphériques décentralisés	676
Tableau 22 – Bit d'état de poids faible de la partie fractionnaire (2^0)	678
Tableau 23 – Plage de valeurs d'état	679
Tableau 24 – OctetString2+Unsigned8	680
Tableau 25 – Octets de Float32+Unsigned8	681
Tableau 26 – Octets de Unsigned8+Unsigned8	681
Tableau 27 – Octets Unsigned16_S	682
Tableau 28 – Signification de Unsigned16_S	682
Tableau 29 – Octets de Integer16_S	682
Tableau 30 – Signification de Integer16_S	682
Tableau 31 – Octets de Unsigned8_S	683
Tableau 32 – Signification de Unsigned8_S	683
Tableau 33 – Octets de OctetString_S	683
Tableau 34 – Bits de statut de OctetString_S	684
Tableau 35 – F message trailer with 4 octets («Queue de message F avec quatre octets»)	684
Tableau 36 – F message trailer with 5 octets («Queue de message F avec cinq octets»)	685
Tableau 37 – Get	693
Tableau 38 – Set	696
Tableau 39 – Local Set Command	700
Tableau 40 – Identify	701
Tableau 41 – Hello	704
Tableau 42 – Start bridge	712
Tableau 43 – Start slave	713
Tableau 44 – Start master	714
Tableau 45 – Stop bridge	715

Tableau 46 – Stop slave	716
Tableau 47 – Stop master	716
Tableau 48 – Sync state change	717
Tableau 49 – Line Delay change	718
Tableau 50 – PPM Set Prov Data	732
Tableau 51 – PPM Set Prov Status	732
Tableau 52 – PPM Activate	733
Tableau 53 – PPM Close	736
Tableau 54 – PPM Start	736
Tableau 55 – PPM Error	736
Tableau 56 – Get Cons Data	737
Tableau 57 – CPM Get cons status	738
Tableau 58 – CPM Set RedRole	739
Tableau 59 – CPM Activate	739
Tableau 60 – CPM NoData	741
Tableau 61 – CPM Stop	742
Tableau 62 – APMS Activate	744
Tableau 63 – APMR Activate	746
Tableau 64 – APMS A Data	747
Tableau 65 – APMR A Data	748
Tableau 66 – APMR Ack	749
Tableau 67 – APMS Error	749
Tableau 68 – APMS Error ERRCLS/ERRCODE	750
Tableau 69 – APMR Error	750
Tableau 70 – APMR Error ERRCLS/ERRCODE	751
Tableau 71 – APMS_Close	751
Tableau 72 – APMR_Close	751
Tableau 73 – Connect	754
Tableau 74 – Release	755
Tableau 75 – Read	756
Tableau 76 – Write	757
Tableau 77 – Control	758
Tableau 78 – System capabilities	763
Tableau 79 – Auto negotiation support and status	765
Tableau 80 – MDI Power Support	766
Tableau 81 – Link aggregation status	766
Tableau 82 – Remote systems data change	769
Tableau 83 – Local Get Time	771
Tableau 84 – Local Set Time	772
Tableau 85 – Local time changed event	772
Tableau 86 – Valeurs permises de Forwarding Mode	776
Tableau 87 – Valeurs permises de Fast Forwarding Multicast MAC Add	776
Tableau 88 – Valeurs permises de Reduction Ratio	778

Tableau 89 – Frame ID	778
Tableau 90 – Tx Port Entry	780
Tableau 91 – Dépendances de RedOrangePeriodBegin, OrangePeriodBegin, et GreenPeriodBegin	783
Tableau 92 – Port state change	786
Tableau 93 – Set port state	786
Tableau 94 – Flush filtering data base	787
Tableau 95 – MAU Type change	792
Tableau 96 – Set MAU Type	792
Tableau 97 – IP Multicast address	796
Tableau 98 – Set ARP Cache	796
Tableau 99 – Enterprise number	799
Tableau 100 – Vendor OUI	800
Tableau 101 – P Data	801
Tableau 102 – N Data	802
Tableau 103 – A Data	804
Tableau 104 – C Data	805
Tableau 105 – Exigences et caractéristiques	806
Tableau 106 – Liaison des services Record Data	831
Tableau 107 – Comportement de la persistance des objets Record Data	835
Tableau 108 – Read	837
Tableau 109 – Services Read (lecture)	839
Tableau 110 – Read Query	841
Tableau 111 – Write	844
Tableau 112 – Services Write (écriture)	845
Tableau 113 – Eléments Data de Write PD Time Data	847
Tableau 114 – Local Write Multiple	848
Tableau 115 – Local New Write Multiple	851
Tableau 116 – Local Set Input	864
Tableau 117 – Local Set Input IOCS	865
Tableau 118 – Local Get Input	866
Tableau 119 – Local Get Input IOCS	867
Tableau 120 – Local New Input	868
Tableau 121 – Local Set Redundancy	869
Tableau 122 – Local Set State	869
Tableau 123 – Local Data State Changed	870
Tableau 124 – Eléments Data du Read Record Input Data Object Element	872
Tableau 125 – Local Set Output	873
Tableau 126 – Local Set Output IOCS	874
Tableau 127 – Local Get Output	875
Tableau 128 – Local Get Output IOCS	876
Tableau 129 – Local New Output	877
Tableau 130 – Local Set Provider State	878

Tableau 131 – Eléments Data du Read Record Output Data Object Element	879
Tableau 132 – Eléments Data du Read Substitute Value	880
Tableau 133 – Eléments Data du Write Substitute Value.....	881
Tableau 134 – Eléments Data de Read LogBook Data	884
Tableau 135 – Local Create LogBook Entry	884
Tableau 136 – Channel Error Type	891
Tableau 137 – Ext Channel Error type	893
Tableau 138 – Combinaisons autorisées de Channel Error Type et Ext Channel Error Type	897
Tableau 139 – Ext Channel Add Value pour Accumulative Info	898
Tableau 140 – Local Add Diagnosis Entry.....	899
Tableau 141 – Local Remove Diagnosis Entry	901
Tableau 142 – Local Diagnosis Event	902
Tableau 143 – Définition générale de Data pour Diagnosis services	904
Tableau 144 – Eléments Data de Read Nested Diagnosis Information	920
Tableau 145 – Primitives distantes émises ou reçues par DIAG_DIAG.....	921
Tableau 146 – Primitives locales émises ou reçues par DIAG_DIAG.....	921
Tableau 147 – Table d'états DIAG_DIAG	923
Tableau 148 – Fonctions, macros, temporisateurs et variables utilisés par DIAG_DIAG.....	923
Tableau 149 – Primitives distantes émises ou reçues par DIAG_MR.....	925
Tableau 150 – Primitives locales émises ou reçues par DIAG_MR.....	925
Tableau 151 – Table d'états DIAG_MR	927
Tableau 152 – Fonctions, macros, temporisateurs et variables utilisés par DIAG_MR.....	927
Tableau 153 – Primitives distantes émises ou reçues par DIAG_MD.....	929
Tableau 154 – Primitives locales émises ou reçues par DIAG_MD.....	929
Tableau 155 – Table d'états DIAG_MD	931
Tableau 156 – Fonctions, macros, temporisateurs et variables utilisés par l'entrée de maintenance exigée.....	931
Tableau 157 – Primitives distantes émises ou reçues par DIAG_QUALIFIED	933
Tableau 158 – Primitives locales émises ou reçues par DIAG_QUALIFIED	933
Tableau 159 – Table d'états DIAG_QUALIFIED	934
Tableau 160 – Fonctions, macros, temporisateurs et variables utilisés par DIAG_QUALIFIED	935
Tableau 161 – Alarm type	938
Tableau 162 – Alarm Notification	942
Tableau 163 – Channel Diagnosis.....	945
Tableau 164 – Manufacturer Specific Diagnosis.....	945
Tableau 165 – Submodule Diagnosis State	945
Tableau 166 – AR Diagnosis State.....	946
Tableau 167 – User Structure Identifier.....	947
Tableau 168 – Sémantique de Spécifier	948
Tableau 169 – Module State	963
Tableau 170 – Usage en rapport avec le type de CR	964
Tableau 171 – Detail.....	965

Tableau 172 – AR Info	966
Tableau 173 – Ident Info	967
Tableau 174 – Liaison des services Context	976
Tableau 175 – Connect	978
Tableau 176 – Connect Device Access	989
Tableau 177 – Release	992
Tableau 178 – Abort	993
Tableau 179 – Local AR Abort	994
Tableau 180 – Prm Begin	995
Tableau 181 – Prm End	997
Tableau 182 – Application Ready	999
Tableau 183 – Définition générale de Data pour les services d'identification	1002
Tableau 184 – Eléments Data de Read Module Diff Block	1005
Tableau 185 – Eléments Data de Read API Data	1007
Tableau 186 – Eléments Data de Read I&M0 Filter Data	1008
Tableau 187 – Eléments Data de Read I&M0 Data	1010
Tableau 188 – Eléments Data de Write I&M1 Data	1011
Tableau 189 – Eléments Data de Read I&M1 Data	1012
Tableau 190 – Eléments Data de Write I&M2 Data	1012
Tableau 191 – Eléments Data de Read I&M2 Data	1013
Tableau 192 – Eléments Data de Write I&M3 Data	1013
Tableau 193 – Eléments Data de Read I&M3 Data	1013
Tableau 194 – Eléments Data de Write I&M4 Data	1014
Tableau 195 – Eléments Data de Read I&M4 Data	1014
Tableau 196 – Eléments Data de Read Autoconfiguration Data	1015
Tableau 197 – Eléments Data de Write Expected Fast Startup Data	1017
Tableau 198 – Eléments Data de Read Expected Fast Startup Data	1018
Tableau 199 – Eléments Data de Read GSD Data	1019
Tableau 200 – Eléments Data de Read GSD Data	1020
Tableau 201 – Eléments Data du Write Isochronous Mode Data	1029
Tableau 202 – Eléments Data de Read Isochronous Mode Data	1030
Tableau 203 – Local SYNCH Event	1030
Tableau 204 – Primitives distantes émises ou reçues par ISOM_SYNC	1033
Tableau 205 – Primitives locales émises ou reçues par ISOM_SYNC	1034
Tableau 206 – Table d'états d'ISOM_SYNC	1034
Tableau 207 – Fonctions, macros, temporisateurs et variables utilisés par ISOM_SYNC ..	1035
Tableau 208 – Primitives distantes qui sont émises ou reçues pour ISOM_OUT	1035
Tableau 209 – Primitives locales qui sont émises ou reçues pour ISOM_OUT	1036
Tableau 210 – Table d'états ISOM_OUT	1038
Tableau 211 – Fonctions, macros, temporisateurs et variables utilisés par ISOM_OUT	1040
Tableau 212 – Primitives distantes émises ou reçues pour ISOM_IN	1041
Tableau 213 – Primitives locales émises ou reçues pour ISOM_IN	1041
Tableau 214 – Table d'états ISOM_IN	1043

Tableau 215 – Fonctions, macros, temporisateurs et variables utilisés par ISOM_IN	1044
Tableau 216 – Numéro de sous-baie pour sous-modules d'interface	1050
Tableau 217 – Sync Properties Role	1054
Tableau 218 – Sync Class	1054
Tableau 219 – Distributed Watchdog Factor.....	1055
Tableau 220 – Restart Factor For Distributed Watchdog	1056
Tableau 221 – DFP Mode	1056
Tableau 222 – SFIOCRProperties.DFPRedundantPathLayout.....	1056
Tableau 223 – SFCRC16	1057
Tableau 224 – Numéro de sous-baie pour sous-modules de port	1059
Tableau 225 – Fiber Optic Type	1060
Tableau 226 – Fiber Optic Cable Type	1060
Tableau 227 – TimePLLWindow.....	1064
Tableau 228 – Eléments Data de Read PDev Data	1064
Tableau 229 – Eléments Data de Read PD Real Data.....	1070
Tableau 230 – Eléments Data de Read PD Expected Data	1073
Tableau 231 - Read PD Interface Data Real	1077
Tableau 232 – Eléments Data de Write PD Interface Adjust.....	1078
Tableau 233 – Eléments Data de Read PD Interface Adjust.....	1079
Tableau 234 – Eléments Data de Write PD IR Data	1079
Tableau 235 – Eléments Data de Read PD IR Data	1083
Tableau 236 – Eléments Data de Write PD Sync Data	1086
Tableau 237 – Eléments Data de Read PD Sync Data	1087
Tableau 238 – Local Sync State Info.....	1088
Tableau 239 – Eléments Data de Write PD IR Subframe Data	1090
Tableau 240 – Eléments Data de Read PD IR Subframe Data	1091
Tableau 241 – Eléments Data de Write PD Time Data	1092
Tableau 242 – Eléments Data de Read PD Time Data	1093
Tableau 243 – Eléments Data de Read PD Interface MRP Data Real.....	1093
Tableau 244 – Eléments Data de Write PD Interface MRP Data Check	1094
Tableau 245 – Eléments Data de Read PD Interface MRP Data Check	1095
Tableau 246 – Eléments Data de Write PD Interface MRP Data Adjust	1095
Tableau 247 – Eléments Data de Read PD Interface MRP Data Adjust	1096
Tableau 248 – Eléments Data de Write PD Interface FSU Data Adjust.....	1097
Tableau 249 – Eléments Data de Read PD Interface FSU Data Adjust.....	1098
Tableau 250 – Eléments Data de Write PD NC Data Check	1099
Tableau 251 – Eléments Data de Read PD NC Data Check	1099
Tableau 252 – Eléments Data de Read PD Port Real Data	1100
Tableau 253 – Eléments Data de Write PD Port Data Check.....	1101
Tableau 254 – Eléments Data de Read PD Port Data Check.....	1102
Tableau 255 – Eléments Data de Write PD Port Data Adjust.....	1102
Tableau 256 – Eléments Data de Read PD Port Data Adjust.....	1103
Tableau 257 – Eléments Data de Read Port FO Data Real	1104

Tableau 258 – Eléments Data de Write PD Port FO Data Check	1105
Tableau 259 – Eléments Data de Read PD Port FO Data Check	1105
Tableau 260 – Eléments Data de Write PD Port FO Data Adjust	1106
Tableau 261 – Eléments Data de Read PD Port FO Data Adjust	1107
Tableau 262 – Eléments Data de Read PD Port MRP Real Data	1107
Tableau 263 – Eléments Data de Write PD Port MRP Data Adjust	1108
Tableau 264 – Eléments Data de Read PD Port MRP Data Adjust	1108
Tableau 265 – Eléments Data de Read PD Port Statistic	1109
Tableau 266 – Device Access	1123
Tableau 267 – Companion AR	1123
Tableau 268 – Acknowledge Companion AR	1124
Tableau 269 – Startup Mode	1124
Tableau 270 – Pull Module Alarm Allowed	1124
Tableau 271 – Input Valid sur Backup AR	1127
Tableau 272 – Activate Redundancy Alarm	1128
Tableau 273 – APStructureIdentifieur avec API := 0	1128
Tableau 274 – APStructureIdentifieur avec API != 0	1129
Tableau 275 – Classes Traffic versus Classe RT	1132
Tableau 276 – Frame ID	1133
Tableau 277 – Reduction Ratios (rapport de réduction)	1134
Tableau 278 – Eléments Data de Read AR Data	1139
Tableau 279 – Local Set AR State	1144
Tableau 280 – Local AR In Data	1144
Tableau 281 – Primitives distantes émises ou reçues par DEVSM	1146
Tableau 282 – Primitives locales émises ou reçues par DEVSM	1147
Tableau 283 – Table d'états DEVSM	1149
Tableau 284 – Fonctions, macros, temporisateurs et variables utilisés par DEVSM	1154
Tableau 285 – Primitives distantes émises ou reçues par REM_CHK	1155
Tableau 286 – Primitives locales émises ou reçues par REM_CHK	1155
Tableau 287 – Table d'états REM_CHK	1156
Tableau 288 – Fonctions, macros, temporisateurs et variables utilisés par REM_CHK	1161
Tableau 289 – Primitives distantes émises ou reçues par LOC_LNK	1162
Tableau 290 – Primitives locales émises ou reçues par LOC_LNK	1162
Tableau 291 – Table d'états de LOC_LNK	1163
Tableau 292 – Fonctions, macros, temporisateurs et variables utilisés par LOC_LNK	1167
Tableau 293 – Primitives distantes émises ou reçues par FOMR	1167
Tableau 294 – Primitives locales émises ou reçues par FOMR	1168
Tableau 295 – Table d'états FOMR	1169
Tableau 296 – Fonctions, macros, temporisateurs et variables utilisés par FOMR	1170
Tableau 297 – Primitives distantes émises ou reçues par FOMD	1170
Tableau 298 – Primitives locales émises ou reçues par FOMD	1170
Tableau 299 – Table d'états FOMD	1171
Tableau 300 – Fonctions, macros, temporisateurs et variables utilisés par FOMD	1172

Tableau 301 – Primitives distantes émises ou reçues par FODIAG	1173
Tableau 302 – Primitives locales émises ou reçues par FODIAG	1173
Tableau 303 – Table d'états FODIAG	1174
Tableau 304 – Fonctions, macros, temporisateurs et variables utilisés par FODIAG	1175
Tableau 305 – Primitives distantes émises ou reçues par RSMSM	1177
Tableau 306 – Primitives locales émises ou reçues par RSMSM	1177
Tableau 307 – Table d'états RSMSM	1178
Tableau 308 – Fonctions, macros, temporisateurs et variables utilisés par RSMSM	1180
Tableau 309 – Règles pour Submodule State.Ident Info	1180
Tableau 310 – Primitives distantes qui sont émises ou reçues par OWNSM et ASSSM	1184
Tableau 311 – Primitives locales qui sont émises ou reçues par OWNSM	1185
Tableau 312 – Table d'états OWNSM	1187
Tableau 313 – Table d'états ASSSM	1194
Tableau 314 – Fonctions, macros, temporisateurs et variables utilisés par OWNSM	1195
Tableau 315 – Fonctions, macros, temporisateurs et variables utilisés par ASSSM	1196
Tableau 316 – Règles pour Submodule State.AR Info	1196
Tableau 317 – Primitives distantes qui sont émises ou reçues par PLUGSM	1197
Tableau 318 – Primitives locales qui sont émises ou reçues par PLUGSM	1198
Tableau 319 – Table d'états PLUGSM	1200
Tableau 320 – Fonctions, macros, temporisateurs et variables utilisés par PLUGSM	1201
Tableau 321 – Primitives distantes qui sont émises ou reçues par PULLSM	1201
Tableau 322 – Primitives locales qui sont émises ou reçues par PULLSM	1202
Tableau 323 – Table d'états PULLSM	1203
Tableau 324 – Fonctions, macros, temporisateurs et variables utilisés par PULLSM	1203
Tableau 325 – Primitives distantes qui sont émises ou reçues par SYNC_DIAG	1204
Tableau 326 – Primitives locales qui sont émises ou reçues par SYNC_DIAG	1204
Tableau 327 – Table d'états SYNC_DIAG	1205
Tableau 328 – Fonctions, macros, temporisateurs et variables utilisés par SYNC_DIAG	1206
Tableau 329 – Primitives distantes qui sont émises ou reçues par CTLSM	1208
Tableau 330 – Primitives locales qui sont émises ou reçues par CTLSM	1208
Tableau 331 – Table d'états CTLSM	1210
Tableau 332 – Fonctions, macros, temporisateurs et variables utilisés par CTLSM	1212
Tableau 333 – Services FAL de l'appareil E/S	1214
Tableau 334 – Services FAL de l'appareil de commande E/S	1215

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**RÉSEAUX DE COMMUNICATION INDUSTRIELS –
SPÉCIFICATIONS DES BUS DE TERRAIN –****Partie 5-10: Définition des services de la couche application –
Éléments de type 10**

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (CEI) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de brevet. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

L'attention est attirée sur le fait que l'utilisation du type de protocole associé est restreinte par les détenteurs des droits de propriété intellectuelle. En tout état de cause, l'engagement de renonciation partielle aux droits de propriété intellectuelle pris par les détenteurs de ces droits autorise l'utilisation d'un type de protocole de couche avec les autres protocoles de couche du même type, ou dans des combinaisons avec d'autres types autorisées explicitement par les détenteurs des droits de propriété intellectuelle pour ce type.

NOTE Les combinaisons de types de protocoles sont spécifiées dans la CEI 61784-1 et la CEI 61784-2.

La Norme internationale CEI 61158-5-10 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette troisième édition annule et remplace la deuxième édition, parue en 2010. Cette édition constitue une révision technique.

Les principaux changements par rapport à l'édition précédente sont les suivants:

- Corrections et améliorations
- Passage de l'intégration MRP à la référence MRP
- Intégration de la condensation dynamique de trames, de leur fragmentation et de leur transfert rapide,
- Intégration de l'autoconfiguration
- Intégration de la redondance de supports sans couture MRPD («Media Redundancy with Planned Duplication» of frames (Redondance de support avec duplication planifiée des trames))
- Intégration de base de la Couche de redondance du système (System Redundancy Layer)
- Intégration de base de la fonctionnalité Configurer pendant le fonctionnement (Configure In Run)
- Optimisation du démarrage et transfert RT_CLASS_3
- Optimisation du temps de démarrage à partir de la coupure d'alimentation électrique
- Nouveau type de câble en fibre GI-PCF
- Suppression de MRRT
- Mise à jour du LLDP-EXT-MIB

Le texte de cette norme est issu des documents suivants.

FDIS	Rapport de vote
65C/763/FDIS	65C/773/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61158, publiées sous le titre général *Réseaux de communication industriels – Spécifications de bus de terrain*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

INTRODUCTION

La présente partie de la CEI 61158 est l'une d'une série produite pour faciliter l'interconnexion de composants d'un système d'automatisation. Elle est liée à d'autres normes de la série telle que définie par le modèle de référence des bus de terrain «à trois couches» décrit dans le rapport CEI 61158-1.

Le service application est fourni par le protocole d'application utilisant les services disponibles de la liaison de données ou autre couche immédiatement inférieure. La présente norme définit les caractéristiques de services d'application pouvant être exploitées par les applications de réseau de terrain et/ou la gestion de système.

Dans toute la série de normes relatives aux bus de terrain, le terme « service » se réfère à la capacité abstraite fournie par une couche du Modèle de référence de base de l'Interconnexion des systèmes ouverts (OSI) à la couche immédiatement supérieure. Ainsi, le service de la couche application défini dans la présente norme est un service architectural conceptuel, indépendant des divisions administratives et de mise en œuvre.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 5-10: Définition des services de la couche application – Eléments de type 10

1 Domaine d'application

1.1 Généralités

La Couche application de réseau de terrain (FAL, Fieldbus Application Layer) fournit des programmes d'utilisateur avec un moyen d'accéder à l'environnement de communication du réseau de terrain. A cet égard, la FAL peut être vue comme une «fenêtre entre des programmes d'application correspondants».

La présente norme fournit les éléments communs pour les communications de messagerie de base à temps critique et à temps non critique entre des programmes d'application dans un environnement d'automation et le matériel spécifique au bus de terrain de Type 10. Le terme « à temps critique » sert à représenter la présence d'une fenêtre temporelle, dans les limites de laquelle une ou plusieurs actions spécifiées sont tenues d'être parachevées avec un certain niveau défini de certitude. Le manquement à parachever les actions spécifiées dans les limites de la fenêtre temporelle risque d'entraîner la défaillance des applications qui demandent ces actions, avec le risque concomitant pour l'équipement, l'installation et éventuellement pour la vie humaine.

La présente norme définit de manière abstraite le service visible de l'extérieur fourni par la couche d'application de bus de terrain de type 10 en termes

- a) d'un modèle abstrait pour définir des ressources (objets) d'application capables d'être manipulées par les utilisateurs par l'intermédiaire de l'utilisation du service FAL;
- b) des actions et événements primitifs du service;
- c) des paramètres associés à chaque action primitive et événement primitif, et la forme qu'ils prennent; et
- d) l'interrelation entre ces actions et événements, et leurs séquences valides.

Le but de la présente norme est de définir les services fournis à

- a) l'utilisateur de FAL à la frontière entre l'utilisateur et la Couche application du Modèle de référence de bus de terrain; et
- b) la Gestion des systèmes au niveau de la frontière entre la Couche application et la Gestion des systèmes selon le Modèle de référence de bus de terrain.

La présente norme spécifie la structure et les services de la couche application des bus de terrain CEI de Type 10, en conformité avec le Modèle de référence de base de l'OSI (ISO/CEI 7498-1) et la structure de la couche application de l'OSI (ISO/CEI 9545).

Les services et protocoles de la FAL sont fournis par des entités d'application (application entity, AE) de la FAL contenues dans les processus d'application. L'AE de la FAL se compose d'un jeu d'éléments de service application (ASE, Application Service Element) orientés objet et d'une entité de gestion de couche (LME, Layer Management Entity) qui gère l'AE. Les ASE fournissent des services de communication qui fonctionnent sur un jeu de classes d'objets de processus application (APO, application process object) connexes. L'un des ASE de la FAL est un ASE de gestion qui fournit un jeu commun de services pour la gestion des instances de classes de la FAL.

Bien que ces services spécifient, du point de vue des applications, la manière dont la demande et les réponses sont émises et délivrées, ils n'incluent pas une spécification de ce que les applications qui demandent et qui répondent doivent en faire. A savoir, les aspects comportementaux des applications ne sont pas spécifiés; seule une définition des demandes et réponses qu'elles peuvent envoyer/recevoir est spécifiée. Cela permet une plus grande flexibilité aux utilisateurs de la FAL pour normaliser un tel comportement d'objet. En plus de ces services, certains services d'appui sont également définis dans la présente norme pour fournir l'accès à la FAL afin de maîtriser certains aspects de son fonctionnement.

1.2 Spécifications

L'objectif principal de la présente norme est de spécifier les caractéristiques des services conceptuels d'une couche application qui sont adaptées à des communications à temps critique et, donc, complètent le Modèle de référence de base de l'OSI en guidant le développement des protocoles de couche application pour les communications à temps critique.

Un objectif secondaire est de fournir des trajets de migration à partir de protocoles de communications industrielles préexistants. C'est ce dernier objectif qui donne naissance à la diversité des services normalisés comme les divers Types de la CEI 61158, et les protocoles correspondants normalisés dans les sous-parties de la CEI 61158-6.

La présente norme peut être utilisée comme la base pour les interfaces de programmation d'applications (Application Programming Interfaces) formelles. Néanmoins, elle n'est pas une interface de programmation formelle et il sera nécessaire pour toute interface de ce type de traiter de questions de mise en œuvre qui ne sont pas couvertes par la présente norme, y compris

- a) les tailles et l'ordonnement des octets pour les divers paramètres de service à plusieurs octets, et
- b) la corrélation de primitives appariées « request-confirm » (c'est-à-dire demande et confirmation) ou « indication-response » (indication et réponse).

1.3 Conformité

La présente norme ne spécifie ni ne contraint les mises en œuvre individuelles ou de produits individuels ni ne contraint les mises en œuvre d'entités de couche application au sein des systèmes d'automation industriels.

Il y a conformité de l'équipement à la présente norme de définition des services de couche application et elle est principalement obtenue par une mise en œuvre du comportement modélisé de l'utilisateur de couche application (par exemple, voir les diagrammes d'états d'utilisateur) accompagnée d'une mise en œuvre de protocoles conformes de couche application qui satisfont aux services de couche application tels que définis dans la présente norme.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

NOTE Toutes les parties de la série CEI 61158, ainsi que la CEI 61784-1 et la CEI 61784-2 font l'objet d'une maintenance simultanée. Les références croisées à ces documents dans le texte se rapportent par conséquent aux éditions datées dans la présente liste de références normatives.

CEI 61158-1:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 1: Présentation et lignes directrices des séries CEI 61158 et CEI 61784*

CEI 61158-5-3:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 5-3: Définition des services de la couche application – Eléments de type 3*

CEI 61158-6-3:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-3: Spécification du protocole de la couche application – Eléments de type 3*

CEI 61158-6-10:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 6-10: Spécification du protocole de la couche application – Eléments de type 10*

CEI 61800-7-203:2008, *Entraînements électriques de puissance à vitesse variable – Partie 7-203: Interface générique et utilisation de profils pour les entraînements électriques de puissance – Spécification du profil de type 3*

CEI 62439-2, *Réseaux industriels de communication – Réseaux de haute disponibilité pour l'automatisation – Partie 2: Protocole de redondance du support (MRP)*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange* (disponible en anglais seulement)

ISO/CEI 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

ISO/CEI 8822, *Technologies de l'information – Interconnexion de systèmes ouverts – Définition du service de présentation*

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

ISO/CEI 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche Application*

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

ISO/CEI 10731, *Technologies de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Conventions pour la définition des services OSI*

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* (disponible en anglais seulement)

ISO 8601, *Éléments de données et formats d'échange – Échange d'information – Représentation de la date et de l'heure*

IEEE 754, *IEEE Standard for Floating-Point Arithmetic*, disponible à l'adresse <<http://www.ieee.org>>

IEEE 802, *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*, disponible à l'adresse <<http://www.ieee.org>>

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

IEEE 802.1D, *IEEE Standard for Local and Metropolitan Area Networks – Media Access Control (MAC) Bridges*, disponible à l'adresse <<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*, disponible à l'adresse <<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*: disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 791, *Internet Protocol*: disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 792, *Internet Control Message Protocol*: disponible à l'adresse <<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*: disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 1034, *Domain names – concepts and facilities*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 1112, *Host Extensions for IP Multicasting*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 1573, *Evolution of the Interfaces Group of MIB-II*; disponible à l'adresse <<http://www.ietf.org>>

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

IETF RFC 2131, *Dynamic Host Configuration Protocol*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 2132, *DHCP Options and BOOTP Vendor Extensions*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 2365, *Administratively Scoped IP Multicast*; disponible à l'adresse <<http://www.ietf.org>>

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

IETF RFC 2737, *Entity MIB (Version 2)*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 2863, *The Interfaces Group MIB*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 3330, *Special-Use IPv4 Addresses*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 3418, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 3490, *Internationalizing Domain Names in Applications (IDNA)*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 3621, *Power Ethernet MIB*; disponible à l'adresse <<http://www.ietf.org>>

IETF RFC 4836, *Definitions of Managed Objects for IEEE 802.3 Medium Attachment Units (MAUs)*, disponible à l'adresse <<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* (disponible à l'adresse <<http://www.opengroup.org/onlinepubs/9629399/toc.htm>>)

3 Termes, définitions, abréviations, symboles et conventions

Pour les besoins du présent document, les termes, définitions, symboles, abréviations et conventions suivants s'appliquent.

3.1 Termes et définitions référencés

3.1.1 Termes de l'ISO/CEI 7498-1

Pour les besoins du présent document, les termes suivants tels que définis dans l'ISO/CEI 7498-1 s'appliquent:

- a) entité d'application
- b) processus d'application
- c) unité de données de protocole application
- d) élément de service application
- e) invocation d'entité d'application
- f) invocation de processus d'application
- g) transaction d'application
- h) système ouvert réel
- i) syntaxe de transfert

3.1.2 Termes de l'ISO/CEI 8822

Pour les besoins du présent document, les termes suivants tels que définis dans l'ISO/CEI 8822 s'appliquent:

- a) syntaxe abstraite
- b) contexte de présentation

3.1.3 Termes de l'ISO/CEI 9545

Pour les besoins du présent document, les termes suivants tels que définis dans l'ISO/CEI 9545 s'appliquent:

- a) application-association -association d'applications)
- b) application-context (contexte d'application())
- c) application context name (nom de contexte d'application)

- d) application-entity-invocation (invocation d'entité d'application)
- e) application-entity-type (type d'entité d'application)
- f) application-process-invocation (invocation de processus d'application)
- g) application-process-type (type de processus d'application)
- h) application-service-element (élément de service application)
- i) application control service element (élément de service de contrôle d'application)

3.1.4 Termes de l'ISO/CEI 8824-1

Pour les besoins du présent document, les termes suivants tels que définis dans l'ISO/CEI 8824-1 s'appliquent:

- a) identificateur d'objet
- b) type

3.2 Termes et définitions complémentaires pour périphérie décentralisée

3.2.1

alarm (alarme)

activation d'un événement qui montre un état critique

3.2.2

alarm ack

acquiescement d'un événement qui montre un état critique

3.2.3

objet de données alarme

objet(s) qui représente(nt) des états critiques référencés par appareil/baie/sous-baie/type d'alarme

3.2.4

allouer

prendre une ressource dans une zone commune et affecter la ressource en question à l'usage exclusif d'une entité spécifique

3.2.5

application

fonction ou structure de données pour laquelle les données sont consommées ou produites

3.2.6

interopérabilité de couche application

capacité des entités d'application d'accomplir des opérations coordonnées et coopératives en utilisant les services de la FAL

3.2.7

objets d'applications

plusieurs classes d'objets qui gèrent et assurent un échange de PDU sur le réseau et dans l'appareil réseau, lors de l'exécution

3.2.8

processus d'application

partie d'une application distribuée sur un réseau, qui est située sur un appareil et adressée sans ambiguïté

3.2.9

identificateur de processus d'application

distingue de multiples processus d'application utilisés dans un appareil

3.2.10

objet de processus d'application

composant d'un processus d'application qui est identifiable et accessible par l'intermédiaire d'une relation entre applications de la FAL

Note 1 à l'article: Les définitions d'objets de processus d'application se composent d'un jeu de valeurs pour les attributs de leur classe (voir la définition de « classe d'objets de processus d'application »). Les définitions d'objet de processus d'application sont accessibles à distance à l'aide des services de l'ASE FAL Management Object (Gestion d'objet de la FAL). Les services de Gestion d'objet de la FAL peuvent être utilisés pour charger ou mettre à jour des définitions d'objet, pour lire des définitions d'objet, et pour créer et supprimer de manière dynamique des objets de processus application et leurs définitions correspondantes.

3.2.11

classe d'objets de processus d'application

classe d'objets de processus d'application définie en termes du jeu de leurs attributs et services accessibles par le réseau

3.2.12

relation entre applications

association coopérative entre deux ou plusieurs invocations d'entités d'application (application-entity-invocation) à des fins d'échange d'informations et de coordination de leur fonctionnement conjoint. Cette relation est activée soit par l'échange d'unités de données de protocole d'application, soit à la suite d'activités de préconfiguration

3.2.13

élément de service d'application de relation entre applications

application-service-element (élément de service d'application) qui fournit le moyen exclusif d'établir et de faire cesser toutes les relations entre applications

3.2.14

point d'extrémité de relation entre applications

contexte et comportement d'une relation entre applications tels que vus et maintenus par l'un des processus d'application impliqués dans la relation entre applications

Note 1 à l'article: Chaque processus d'application impliqué dans la relation entre applications maintient son propre point d'extrémité de relation entre applications

3.2.15

attribut

description d'une caractéristique visible de l'extérieur d'un objet

Note 1 à l'article: Les attributs d'un objet contiennent des informations sur les portions variables d'un objet. Typiquement, ils fournissent des informations relatives au statut ou régissent le fonctionnement d'un objet. Des attributs peuvent aussi avoir une incidence sur le comportement d'un objet. Les attributs se répartissent en attributs de classes et attributs d'instances.

3.2.16

backup (secours)

statut de l'IO AR (AR E/S), qui indique qu'elle est dans l'état standby (d'attente)

3.2.17

comportement

indication de la façon dont un objet réagit à des événements particuliers

3.2.18

voie

représentation d'une simple liaison physique ou logique d'un objet d'application d'entrée ou de sortie d'un serveur au processus afin de prendre en charge l'adressage d'informations relatives au diagnostic

Note 1 à l'article: La voie représente typiquement un simple connecteur ou une simple pince comme étant une interface réelle d'un module ou sous-module. Cette référence est utilisée pour identifier des points de défaillance au sein des PDU de diagnostic.

3.2.19

diagnostic lié à une voie

informations concernant un élément spécifique d'un objet de processus d'application d'entrée ou de sortie, fournies à des fins de maintenance

EXEMPLE boucle ouverte

3.2.20

classe

ensemble d'objets, qui représentent tous le même type de composant système

Note 1 à l'article: Une classe est une généralisation d'un objet; un modèle pour définir des variables et des méthodes. Tous les objets dans une classe ont une forme et un comportement identiques, mais contiennent en général des données différentes dans leurs attributs.

3.2.21

attributs de classe

attribut qui est partagé par tous les objets au sein de la même classe

3.2.22

code de classe

identificateur unique attribué à chaque classe d'objets

3.2.23

service spécifique à une classe

service défini par une classe d'objets particulière pour accomplir une fonction requise qui n'est pas accomplie par un service commun

Note 1 à l'article: Un objet spécifique à une classe est unique pour la classe d'objets qui le définit.

3.2.24

dégagé

statut de l'appareil de commande E/S qui indique que l'algorithme de commande n'est pas en cours d'exécution

3.2.25

client

- a) objet qui utilise les services d'un autre objet (serveur) pour accomplir une tâche
- b) initiateur d'une PDU à laquelle un serveur réagit

3.2.26

profil commun

ensemble d'informations et de fonctionnalité indépendantes vis-à-vis de tout appareil qui assure la cohérence entre tous les appareils

3.2.27

objet de données de communication

objet(s) qui est/sont paramètre(s) de relations de communication et référencé(s) par appareil/baie/sous-baie/indice

3.2.28

vérification de configuration

comparaison de la structuration attendue de l'objet de données E/S côté client avec la structuration effective de l'objet de données E/S côté serveur dans la phase de démarrage

3.2.29

défaut de configuration

différence inacceptable entre la structuration attendue de l'objet de données E/S et la structuration effective de l'objet de données E/S, telle que détectée par le serveur

3.2.30

identificateur de configuration

représentation d'une partie des données E/S d'un seul module d'entrée et/ou de sortie d'un serveur

3.2.31

consommer

recevoir des données provenant d'un fournisseur

3.2.32

consommateur

nœud ou puits recevant des données provenant d'un fournisseur

3.2.33

gestion de contexte

information accessible par le réseau (objets de communication) qui appuie la gestion du fonctionnement du système de bus de terrain, y compris la couche application

Note 1 à l'article: La gestion inclut les fonctions telles que la commande, la surveillance et le diagnostic.

3.2.34

trajet d'acheminement

flux unidirectionnel d'unités APDU à travers une relation entre applications

3.2.35

cyclique

répétitif d'une manière régulière

3.2.36

cohérence de données

moyen pour une émission et un accès cohérents de l'objet de donnée d'entrée ou de sortie entre client et serveur et au sein du client et du serveur

3.2.37

appareil

équipement matériel physique relié à la liaison

NOTE 1 à l'article: Un appareil peut contenir plus d'un nœud.

3.2.38

ID d'appareil

identification de type d'appareil attribuée par le fournisseur

3.2.39

profil d'appareil

ensemble d'informations et de fonctionnalité dépendant de l'appareil qui assure la cohérence entre des appareils similaires relevant du même type d'appareil

3.2.40

objet de donnée de diagnostic

objet(s) qui contien(nen)t des informations de diagnostic référencées par appareil/baie/sous-baie/indice

3.2.41**information de diagnostic**

toute donnée disponible au niveau du serveur à des fins de maintenance

3.2.42**reconfiguration dynamique**

changement d'objets de données E/S sans interruption d'une relation entre applications établie et mise à jour continue d'objets de données E/S inchangés

3.2.43**point d'extrémité**

l'une des entités en communication impliquées dans une connexion

3.2.44**ingénierie**

terme abstrait qui caractérise l'application ou l'appareil côté client qui est chargé(e) de configurer un système d'automatisation

3.2.45**erreur**

discordance entre une valeur ou un état calculé(e), observé(e) ou mesuré(e) et la valeur ou l'état spécifié(e) ou théoriquement correct(e)

3.2.46**classe d'erreurs**

regroupement général pour des définitions d'erreurs connexes et des codes d'erreurs correspondants

3.2.47**code d'erreur**

identification d'un type spécifique d'erreur dans une classe d'erreurs

3.2.48**événement**

instance d'un changement de conditions

3.2.49**diagnostic étendu relatif à la voie**

informations concernant un élément spécifique d'un objet de processus d'application spécifique, fournies à des fins de maintenance

EXEMPLE Link Fail (défaillance de liaison)

3.2.50**démarrage rapide**

procédure accélérée pour installer un appareil E/S après un cycle de puissance

3.2.51**trame**

unité de transfert de données sur une liaison

3.2.52**objet de donnée d'identification**

objet(s) qui contien(nen)t des informations relatives au fabricant et au type d'appareil, de module et de sous-module référencé(s) par appareil/baie/sous-baie/indice

3.2.53

point d'extrémité d'AR implicite

point d'extrémité d'AR qui est défini localement dans l'appareil sans utilisation du service « create » (création)

3.2.54

indice

adresse d'un objet de données d'enregistrement au sein d'un processus d'application

3.2.55

instance

occurrence physique réelle d'un objet au sein d'une classe qui identifie l'un des plusieurs objets au sein de la même classe d'objets

3.2.56

attribut d'instance

attribut qui est propre à une instance d'objet et n'est pas partagé par la classe d'objets

3.2.57

instancié

objet qui a été créé dans un appareil

3.2.58

invocation

acte consistant à utiliser un service ou une autre ressource d'un processus d'application

Note 1 à l'article: Chaque invocation représente un processus distinct de commande qui peut être décrit par son contexte. Une fois que le service s'achève ou que l'utilisation de la ressource est libérée, l'invocation cesse d'exister. Pour des invocations de service, un service a été lancé mais n'est pas encore achevé s'appelle invocation de service en cours. Egalement, pour des invocations de service, un « Invoke ID » peut être utilisé pour identifier sans ambiguïté l'invocation de service et la différencier d'autres invocations de service en cours.

3.2.59

appareil de commande E/S

appareil de commande qui agit comme client pour plusieurs appareils E/S (appareils de terrain)

Note 1 à l'article: Il s'agit habituellement d'un appareil de commande (« automate ») programmable ou d'un système de commande distribué.

3.2.60

objet de données E/S

objet désigné pour être transféré de façon cyclique pour le besoin de traitement et référencé par appareil/baie/sous-baie

3.2.61

appareil E/S

appareil de terrain qui agit comme serveur pour l'opération E/S

3.2.62

serveur de paramètre E/S

serveur pour paramètre d'application d'appareils E/S (clients)

Note 1 à l'article: Il s'agit habituellement d'un appareil pour effectuer des copies de secours des données de paramètre et pour consigner dans un journal les changements en ligne du paramètre de l'appareil.

3.2.63

sous-système E/S

sous-système constitué d'un appareil de commande E/S et de tous ses appareils E/S associés

3.2.64**superviseur E/S**

appareil d'ingénierie qui gère la mise en service et le diagnostic d'un système E/S

3.2.65**système E/S**

système constitué de tous ses sous-systèmes E/S

Note 1 à l'article: A titre d'exemple, un automate programmable ayant plus d'un appareil de commande E/S (interface réseau) commande un système E/S constitué d'un sous-système E/S pour chaque appareil de commande E/S.

3.2.66**mode isochrone**

mode dans lequel un système E/S fonctionne en étant étroitement synchronisé avec une gigue de moins de 1 μ s

3.2.67**membre**

élément d'un attribut qui est structuré comme étant un élément d'une matrice

3.2.68**méthode**

synonyme pour un service opérationnel qui est fourni par l'ASE serveur et invoqué par un client

3.2.69**module**

composant matériel ou logiciel d'un appareil physique

3.2.70**réseau**

ensemble de nœuds reliés par un certain type de support de communication, notamment des répéteurs intermédiaires, ponts, routeurs et passerelles de couche inférieure

3.2.71**objet**

représentation abstraite d'un composant particulier au sein d'un appareil, habituellement un ensemble de données connexes (sous la forme de variables) et de méthodes (procédures) pour effectuer des opérations sur les données en question qui ont une interface et un comportement clairement définis

3.2.72**service spécifique à un objet**

service propre à la classe d'objets qui le définit

3.2.73**operate («fonctionnement»)**

statut de l'appareil de commande E/S qui indique que l'algorithme de commande est actuellement en cours d'exécution

3.2.74**peer (homologue/pair)**

rôle d'un point d'extrémité d'AR dans lequel il est capable d'agir tant comme client que comme serveur

3.2.75**appareil physique**

appareil d'automatisation ou autre appareil de réseau

3.2.76

connexion point à point

connexion qui existe entre exactement deux objets de processus d'application

3.2.77

primary

statut de l'IO AR («AR E/S»), qui indique qu'elle est dans l'état de fonctionnement

Note 1 à l'article: Outre une primary IO AR («AR E/S primaire»), une backup IO AR («AR E/S de secours») peut exister. Par exemple utilisé pour la redondance et la reconfiguration dynamique de données E/S.

3.2.78

fournisseur

nœud ou source envoyant des données à un ou plusieurs consommateurs

3.2.79

domaine PTCP

un certain nombre de sous-domaines PTCP dans un sous-réseau IP

3.2.80

sous-domaine PTCP

certaines quantités de DTE («équipements terminaux de traitement de données»(ETTD)) avec des horloges synchronisées

3.2.81

objet de données d'enregistrement

objet(s) qui est/sont déjà prétraité(s) et transféré(s) de manière acyclique à des fins d'informations ou de traitement supplémentaire et référencé(s) par appareil/baie/sous-baie/indice

3.2.82

ressource

capacité de traitement ou d'informations

3.2.83

run («marche»)

statut de l'appareil de commande E/S qui indique que l'algorithme de commande est actuellement opérant

3.2.84

serveur

- a) rôle d'un AREP dans lequel il retourne une APDU de réponse de service confirmé au client qui a émis la demande
- b) objet qui fournit des services à un autre objet (client)

3.2.85

service

opération ou fonction qu'un objet et/ou une classe d'objets accompli(ssen)t à la demande d'un autre objet et/ou d'une autre classe d'objets

3.2.86

intervalle de temps

adresse d'une unité de structure au sein d'un appareil E/S

Note 1 à l'article: Au sein d'un appareil modulaire, une baie adresse typiquement un module physique. Au sein d'appareils compacts, une baie adresse typiquement une fonction logique ou un module virtuel.

3.2.87**stop («arrêt»)**

statut de l'appareil de commande E/S qui indique que l'algorithme de commande n'est pas en cours d'exécution

3.2.88**sous-module**

composant matériel ou logiciel d'un module

3.2.89**sous-baie**

adresse d'une unité structurelle au sein d'une baie

Note 1 à l'article: Une sous-baie peut adresser une interface physique pour des sous-modules au sein d'un module. D'une manière générale, une sous-baie est un second niveau aux données de structure au sein d'un appareil.

3.2.90**vendor ID (ID de vendeur)**

numéro administratif central utilisé comme identification de fabricant

3.3 Termes et définitions complémentaires pour la redondance des supports**3.3.1****défaillance**

cessation de l'aptitude d'une entité à accomplir une fonction requise

Note 1 à l'article: Après défaillance d'une entité, cette entité est en état de panne.

Note 2 à l'article: Une « défaillance » est un passage d'un état à un autre, par opposition à une « panne », qui est un état.

Note 3 à l'article: La notion de défaillance, telle qu'elle est définie, ne s'applique pas à une entité constituée seulement de logiciel.

3.3.2**panne**

état d'une entité inapte à accomplir une fonction requise, non compris l'inaptitude due à la maintenance préventive ou à d'autres actions programmées ou due à un manque de moyens extérieurs

Note 1 à l'article: Une panne est souvent la conséquence d'une défaillance de l'entité elle-même, mais elle peut exister sans défaillance préalable.

3.3.3**rétablissement**

récupération de l'aptitude d'une entité à accomplir une fonction requise après une panne

3.3.4**temps de rétablissement**

durée requise pour le rétablissement

3.3.5**redondance**

existence, dans une entité, de plus d'un moyen pour accomplir une fonction requise

Note 1 à l'article: Dans ce contexte, l'existence de plus d'un chemin (consistant en liens et commutateurs) entre nœuds d'extrémité.

3.3.6**anneau**

réseau où chaque nœud est relié en série à deux autres nœuds

Note 1 à l'article: Les nœuds sont reliés les uns aux autres sous la forme logique d'un cercle.

Note 2 à l'article: Les trames sont transmises séquentiellement entre des nœuds actifs, capables chacun d'examiner ou de modifier la trame avant de la transmettre.

3.3.7

port d'anneau

port d'un commutateur auquel une liaison d'anneau est attachée

3.3.8

liaison d'anneau

liaison qui relie deux commutateurs d'un anneau

3.3.9

nœud commutateur

pont MAC tel que défini dans l'IEEE 802.1D

Note 1 à l'article: Il est appelé « commutateur » dans le présent contexte.

3.4 Abréviations et symboles

3.4.1 Généralités

AE	Application Entity (Entité d'application)
AL	Application Layer (Couche application)
ALME	Application Layer Management Entity (Entité de gestion de couche application)
ALP	Application Layer Protocol (Protocole de couche application)
AP	Application Process (Processus d'application)
APDU	Application Protocol Data Unit (Unité de donnée de protocole application)
API	Application Process Identifier (Identificateur de processus d'application)
APMR	Acyclic Protocol machine Receiver (Destinataire de diagramme de protocole acyclique)
APMS	Acyclic Protocol machine Sender (Expéditeur de diagramme de protocole acyclique)
APO	Application Process Object (Objet de processus d'application)
AR	Application Relationship (Relation entre applications)
AREP	Application Relationship End Point (Point d'extrémité de relation entre applications)
ARL	Application Relationship List (Liste de relations entre applications)
ASCII	American Standard Code for Information Interchange (Code américain normalisé pour l'échange d'information)
ASE	Application Service Element (Élément de service application)
BMA	Best Master Algorithm (Algorithme de meilleure mère/meilleur maître)
CID	Connection ID (ID de connexion)
CIM	Computer Integrated Manufacturing (Productique)
CIP	Control and Information Protocol (Protocole de commande et d'information)
CR	Communication Relationship (Relation de communication)
CREP	Communication Relationship End Point (Point d'extrémité de relation de communication)
CTLSM	Controller state machine (Diagramme d'états d'appareil de commande)
DEVSM	Device state machine (Diagramme d'états d'appareil)
DIAG_DIAG	Diagramme d'états d'entrée de diagnostic
DIAG_MD	Diagramme d'états d'entrée demandé par la maintenance
DIAG_MR	Diagramme d'états d'entrée requis par la maintenance
DIAG_QUALIFIED	Diagramme d'états d'entrée qualifié
DL-	préfixe relatif à la liaison de données
DLC	Data Link Connection (Connexion de liaison de données)
DLL	Data Link Layer (Couche liaison de données)
DLSAP	Data link Service Access Point (Point d'accès au service liaison de données)
DNS	Domain Name Service (Service de noms de domaine)
DP	Decentralised Peripherals (Périphériques décentralisés)

FAL	Fieldbus Application Layer (Couche application de bus de terrain)
FIFO	First-In First-Out (Premier entré, premier sorti)
FODIAG	Fiber Optic Diagnosis (Diagnostic de fibre optique)
FOMD	Fiber Optic Maintenance Demanded (Maintenance fibre optique exigée)
FOMR	Fiber Optic Maintenance Required (Maintenance fibre optique requise)
IHM (HMI en anglais)	Interface Homme Machine (en anglais: Human machine interface)
ID	Identificateur
CEI	Commission Electrotechnique Internationale
IP	Internet Protocol (Protocole Internet)
ISO	Organisation internationale de normalisation
LME	Layer Management Entity (Entité de gestion de couche)
LOC_LNK	Local Link (Liaison locale)
MAU	Medium Attachment Unit (Unité de raccordement au support)
MIB	Management Information Base (Base de données MIB)
MRC	Media Redundancy Client (Client de redondance de supports)
MRM	Media Redundancy Manager (Gestionnaire de redondance de supports)
MRP	Media Redundancy Protocol (Protocole de redondance de supports)
MRPD	Media Redundancy (Redondance de support) avec Planned Duplication (Duplication planifiée) de trames
OSI	Open Systems Interconnect (Interconnexion de systèmes ouverts)
OWNSM	Ownership Submodule (Sous-module de propriété)
PDev	Physical Device (Appareil physique)
PDU	Protocol Data Unit (Unité de données de protocole)
PL	Physical Layer (Couche physique)
PLUGSM	Plug State Machine (Diagramme d'états de branchement)
PULLSM	Pull State Machine (Diagramme d'états de retrait)
QoS	Quality of Service (Qualité de service)
REM_CHK	Remote check (Vérification distante) (vérification de proximité)
REP	Route Endpoint (Point d'extrémité de chemin)
RSMSM	Real Submodule State Machine (Diagramme d'états de sous-module réel)
SAP	Service Access Point (Point d'accès au service)
SCL	Security Level (Niveau de sécurité)
SDU	Service Data Unit (Unité de donnée de service)
SEM	State Event Matrix (Matrice d'événements d'états)
SMIB	System Management Information Base (Base d'informations de gestion de système)
SMK	System Management Kernel (Noyau de gestion système)
SNMP	Simple Network Management Protocol (protocole simple de gestion de réseau)
STD	State Transition Diagram (Diagramme des transitions d'états), utilisé pour décrire le comportement d'objet
S-VFD	Simple Virtual Field Device (Appareil de terrain virtuel simple)
SYNC_DIAG	Synchronisation Diagnosis (Diagnostic de synchronisation)
UTC (TUC)	Universal Time Coordinated (Temps universel coordonné)

3.4.2 Abréviations et symboles complémentaires pour périphérie décentralisée

ALPMI	Alarm Protocol Machine Initiator (Initiateur de diagramme de protocole d'alarme)
ALPMR	Alarm Protocol Machine Responder (Répondeur de diagramme de protocole d'alarme)
ARP	Address Resolution Protocol (Protocole de résolution d'adresse)
BBUB	Buffer Buffer Unconfirmed Bidirectional, (Tampon Tampon Non Confirmé bidirectionnel)
BBUU	Buffer Buffer Unconfirmed Unidirectional (Tampon Tampon Non Confirmé unidirectionnel)
BBUU-OM	Buffer Buffer Unconfirmed Unidirectional - One to Many, (Tampon Tampon Non Confirmé Unidirectionnel - Un à Plusieurs)
BMC	Best Master Clock (Meilleure horloge-mère)
CL-RPC	Connectionless Remote Procedure Call (Appel de procédure distante sans connexion)
CM	Context Management (Gestion de contexte)
CPM	Cyclic Protocol Machine (Diagramme de protocole cyclique)
DCE	OSF Distributed Computing Environment (Environnement informatique distribué de l'OSF (Open Software Foundation))
DCP	Discovery and basic Configuration Protocol (Protocole de découverte et de configuration de base)
DHCP	Dynamic Host Configuration Protocol (Protocole de configuration dynamique d'hôte)
DIM	Device Interface Module (Module d'interface d'appareil)
DLPD	Data-Link-Protocol Data Unit (Unité de donnée de protocole de liaison de données)
DTE	Data Terminal Equipment (Équipement terminal de traitement de données (ETTD))
GSDML	General Station Description Markup Language (Langage de balisage de description de station)
I&M	Profil d'identification et de maintenance
IANA	Internet Assigned Numbers Authority (Autorité chargée de l'assignation des numéros Internet)
ICMP	Internet Control Message Protocol (Protocole de message de contrôle sur Internet)
IOCR	Input Output Communication Relation (Relation de communication entrée/sortie)
IOCS	Input Output Object Consumer Status (Statut de consommateur d'objets d'entrée/sortie)
IOPS	Input Output Object Provider Status (Statut de fournisseur d'objets d'entrée/sortie)
IR	Isynchronous Relay (Relais isochrone)
IRT	Isynchronous Real Time Protocol (Protocole temps réel isochrone)
ISO	Organisation internationale de normalisation
Isom	Isynchronous Mode (Mode isochrone)
ISOM_IN	Isynchronous Mode Input Data state machine (Diagramme d'états de données d'entrée en mode isochrone)
ISOM_OUT	Isynchronous Mode Output Data state machine (Diagramme d'états de données de sortie en mode isochrone)
ISOM_SYNC	Isynchronous Mode Sync state machine (Diagramme d'états de synchronisation en mode isochrone)
LED	Light-Emitting Diode (Diode électroluminescente (DEL))
LLDP	Link Layer Discovery Protocol (Protocole de découverte de couche liaison)
LME	Layer Management Entity (Entité de gestion de couche)
lsb	Least Significant Bit (Bit de poids faible)
LT	Longueur/Type
MAC	Medium access control (Commande d'accès au support)
MCR	Multicast Communication Relation (Relation de communication multidiffusion)
msb	Most Significant Bit (Bit de poids fort)
NCA	Network Computing Architecture (Architecture de calcul réseau)

PL	Physical Layer (Couche physique)
PPM	Cyclic Provider Protocol Machine (Diagramme de protocole cyclique de fournisseur)
PTCP	Precision Transparent Clock Protocol (Protocole d'horloge transparent de précision)
QQCB-CL	Queue Queue Confirmed Bi-directional - Connectionless, (File d'attente File d'attente Confirmé Bidirectionnel - Sans connexion)
QQCB-CO	Queue Queue Confirmed Bi-directional - Connection-Oriented, (File d'attente File d'attente Confirmé Bidirectionnel - orienté connexion)
RPC	Remote Procedure Call (Appel de procédure distante)
RT	Real Time (Temps réel)
RTA	Real Time Protocol Acyclic (Protocole temps réel - acyclique)
RTC	Real Time Protocol Cyclic (Protocole temps réel - cyclique)
RTC3PSM	Realtime Class 3 Port State Machine (Diagramme d'états du port 3 de classe Temps réel)
TLV	Type Length Value (Type/Longueur/Valeur) (règle de codage)
UDP	User Datagram Protocol (Protocole datagramme d'utilisateur)
UUID	Universally Unique Identifier (Identificateur unique universel)
VLAN	Virtual Local Area Network (Réseau local virtuel)

3.4.3 Abréviations et symboles pour les services

cnf	Confirmation
ind	Indication
req	Request
rsp	Response

3.5 Conventions

3.5.1 Vue d'ensemble

La couche FAL est définie comme un jeu d'éléments ASE orientés objet. Chaque ASE est spécifié dans un paragraphe distinct. Chaque spécification d'ASE est constituée de deux parties, à savoir sa spécification de classe et sa spécification de service.

La spécification de classe définit les attributs de la classe. Les attributs sont accessibles à partir d'instances de la classe en utilisant les services d'ASE de gestion d'objets spécifiés à l'Article 5 de la présente norme. La spécification de services définit les services qui sont fournis par l'ASE.

3.5.2 Conventions générales

La présente norme utilise les conventions descriptives données dans l'ISO/CEI 10731.

3.5.3 Conventions pour les définitions de classes

Les définitions de classes sont décrites à l'aide de modèles. Chaque modèle est constitué d'une liste d'attributs de la classe. La forme générale du modèle est montrée ci-dessous:

FAL ASE:	Nom de l'ASE
CLASS:	Nom de la classe
CLASS ID:	#
PARENT CLASS:	Nom de la classe parent
ATTRIBUTS:	
1	(o) Attribut clé: identificateur numérique
2	(o) Attribut clé: nom
3	(m) Attribut: nom d'attribut (valeurs)
4	(m) Attribut: nom d'attribut (valeurs)
4.1	(s) Attribut: nom d'attribut (valeurs)

4.2	(s) Attribut:	nom d'attribut (valeurs)
4.3	(s) Attribut:	nom d'attribut (valeurs)
5	(c) Contrainte:	expression de la contrainte
5.1	(m) Attribut:	nom d'attribut (valeurs)
5.2	(o) Attribut:	nom d'attribut (valeurs)
6	(m) Attribut:	nom d'attribut (valeurs)
6.1	(s) Attribut:	nom d'attribut (valeurs)
6.2	(s) Attribut:	nom d'attribut (valeurs)

SERVICES:

1	(o) OpsService:	nom du service
2	(c) Contrainte:	expression de la contrainte
2.1	(o) OpsService:	nom de service
3	(m) MgtService:	nom de service

- (1) La rubrique "FAL ASE:" est le nom de l'élément ASE de la couche FAL (FAL ASE) qui fournit les services pour la classe spécifiée.
- (2) La rubrique "CLASS:" est le nom de la classe spécifiée. Tous les objets définis à l'aide de ce modèle seront une instance de cette classe. La classe peut être spécifiée par la présente norme ou par un utilisateur de la présente norme.
- (3) La rubrique "CLASS ID:" est un numéro qui identifie la classe spécifiée. Ce numéro est unique au sein du FAL ASE qui fournira les services pour cette classe. Lorsqu'il est qualifié par l'identité de son FAL ASE, il identifie sans ambiguïté la classe relevant du domaine d'application de la FAL. La valeur "NULL" indique que la classe ne peut pas être instanciée. Les Class ID (identificateurs de classe) entre 1 et 255 sont réservés par la présente norme pour identifier des classes normalisées. Ils ont été attribués pour conserver la compatibilité avec des normes nationales existantes. Les CLASS ID entre 256 et 2 048 sont alloués pour identifier les classes définies par l'utilisateur.
- (4) La rubrique "PARENT CLASS:" est le nom de la classe parent pour la classe spécifiée. Tous les attributs définis pour la classe parent et hérités par celle-ci sont hérités pour la classe définie, et par conséquent ils ne doivent pas être redéfinis dans le modèle pour cette classe.

NOTE La classe parente "TOP" indique que la classe définie est une définition de classe initiale. La classe parent "TOP" est utilisée comme point de départ à partir duquel toutes les autres classes sont définies. L'usage de "TOP" est réservé pour les classes définies par la présente norme.

- (5) L'étiquette "ATTRIBUTES" (Attributs) indique que les entrées suivantes sont des attributs définis pour la classe.
 - a) Chacune des entrées d'attribut contient un numéro de ligne dans la colonne 1, un indicateur obligatoire (m) / facultatif (o) / conditionnel (c) / sélecteur (s) dans la colonne 2, une étiquette de type d'attribut dans la colonne 3, un nom ou une expression conditionnelle dans la colonne 4, et, facultativement, une liste de valeurs énumérées dans la colonne 5. Dans la colonne suivant la liste de valeurs, la valeur par défaut pour l'attribut peut être spécifiée.
 - b) Les objets sont normalement identifiés par un identificateur numérique et/ou par un nom d'objet. Dans les modèles de classe, ces attributs clés sont définis sous l'attribut clé.
 - c) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication est utilisée pour spécifier
 - i) des champs d'un attribut structuré (4.1, 4.2, 4.3),
 - ii) des attributs conditionnés à un énoncé de contrainte (5). Les attributs peuvent être obligatoires (5.1) ou facultatifs (5.2) si la contrainte est vraie. Tous les attributs facultatifs n'exigent pas des énoncés de contraintes comme le fait l'attribut défini en (5.2),
 - iii) les champs sélection d'un attribut de type choix (6.1 et 6.2).

- (6) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe.
- a) Un (m) dans la colonne 2 indique que le service est obligatoire pour la classe, alors qu'un (o) indique qu'il est facultatif. Un (c) dans cette colonne indique que le service est conditionnel. Lorsque tous les services définis pour une classe le sont comme étant facultatifs, l'un au moins doit être sélectionné quand une instance de la classe est définie.
 - b) L'étiquette "OpsService" désigne un service opérationnel (1).
 - c) L'étiquette "MgtService" désigne un service de gestion (2).
 - d) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication dans la liste de services sert à spécifier des services conditionnés à un énoncé de contrainte.

3.5.4 Conventions pour les définitions des services

3.5.4.1 Généralités

Le modèle de service, les primitives de service et les diagrammes de séquence temporelle utilisés sont des descriptions totalement abstraites; ils ne constituent pas une spécification pour une mise en œuvre.

3.5.4.2 Paramètres de service

Les primitives de service sont utilisées pour représenter les interactions entre utilisateur de service et fournisseur de service (ISO/CEI 10731). Elles acheminent des paramètres qui indiquent des informations disponibles dans l'interaction entre utilisateur et fournisseur. Dans n'importe quelle interface particulière, il n'est pas indispensable d'énoncer tous les paramètres de façon explicite.

Les spécifications de service selon la présente norme utilisent un format de tableau pour décrire les paramètres de composants des primitives du service d'ASE. Les paramètres qui s'appliquent à chaque groupe de primitives de service sont consignés en tableaux. Chaque tableau comporte jusqu'à cinq colonnes pour le/la:

- 1) Nom de paramètre,
- 2) primitive "request",
- 3) primitive "indication",
- 4) primitive "response", et
- 5) primitive "confirm".

Un paramètre (ou un composant de celui-ci) est énuméré dans chaque rangée de chaque tableau. Dans les colonnes appropriées de la primitive de service, un code est utilisé pour spécifier le type d'usage du paramètre sur la primitive spécifiée dans la colonne:

- M Le paramètre est obligatoire pour la primitive.
- U Le paramètre est une option de l'utilisateur et peut ou peut ne pas être fourni, cela dépendant de l'usage dynamique de l'utilisateur du service. Cela ne signifie pas que la prise en charge du paramètre est facultative. Lorsqu'il n'est pas fourni, une valeur par défaut est supposée pour le paramètre;
- C Le paramètre est conditionné à d'autres paramètres ou à l'environnement de l'utilisateur du service;
 - (blanc/vidé) le paramètre n'est jamais présent.
- S Le paramètre est un élément sélectionné.

Certaines entrées sont en plus qualifiées par des éléments entre parenthèses. Ceux-ci peuvent être

- a) une contrainte spécifique au paramètre:
 "(=)" indique que le paramètre équivaut du point de vue de la sémantique au paramètre dans la primitive de service située immédiatement à sa gauche dans le tableau;
- b) une indication qu'une certaine note s'applique à l'entrée;
 "(n)" indique que la note "n" suivante contient des informations complémentaires relatives au paramètre et à son utilisation.

3.5.4.3 Procédures de service

Les procédures sont définies en termes des

- interactions entre entités d'application par l'échange d'unités de données de protocole d'application de bus de terrain, et
- interactions entre un fournisseur de services de couche application et un utilisateur de service de couche application dans le même système par l'invocation de primitives de service de couche application.

Ces procédures sont applicables à des instances de communication entre systèmes qui prennent en charge des services de communications à contrainte temporelle au sein de la couche application de bus de terrain.

3.5.5 Conventions utilisées dans les diagrammes d'états

Les séquences de protocoles sont décrites au moyen de diagrammes d'états.

Dans les diagrammes d'états, les états sont représentés sous forme de boîtes, tandis que les transitions d'états sont montrées comme des flèches. Les noms des états et des transitions du diagramme d'états correspondent aux noms dans l'énumération textuelle des transitions d'états.

L'initialisation de tous les paramètres (des valeurs par défaut, par exemple) d'un diagramme d'états est réalisée automatiquement lors de l'instanciation du diagramme (nouvelle instance). Transition implicite entre «POWER-ON» (mise sous tension) et le premier état appelant le service INIT (initialisation). S'il est nécessaire d'avoir différentes instances d'un type de diagramme, différents paramètres sont définis par le service INIT.

L'énumération textuelle des transitions d'états est structurée comme suit (voir aussi le Tableau 1).

- La première rangée contient le nom de la transition.
- La deuxième rangée définit l'état actuel.
- La troisième rangée contient un événement facultatif suivi de Conditions commençant par une barre oblique "/" comme premier caractère de ligne et, pour finir, suivi par les Actions commençant par un "=>" comme premier caractère de ligne.
- La dernière rangée contient le prochain état.

Si l'événement survient et les conditions sont remplies, la transition se déclenche, c'est-à-dire que les actions sont entreprises et il se produit le passage à l'état suivant.

La présentation d'un diagramme est montrée dans le Tableau 1. La signification des éléments de description d'un diagramme d'états est représentée dans le Tableau 2.

Tableau 1 – Eléments de description de diagramme d'états

#	Etat courant	Evénement ou condition => action	Etat suivant

Tableau 2 – Description d'éléments de diagramme d'états

Elément de description	Signification
Etat courant	Nom des états donnés
Etat suivant	
#	Nom ou numéro de la transition d'état.
Evénement	Nom ou description de l'événement.
/Condition	Expression booléenne. Le symbole "/" qui précède ne fait pas partie de la condition.
=> Action	Liste des assignations et invocations de service ou de fonction. Le symbole "=>" qui précède ne fait pas partie de l'action.

Les conventions utilisées dans les diagrammes d'états sont montrées dans le Tableau 3.

Tableau 3 – Conventions utilisées dans les diagrammes d'états

Convention	Signification
:=	La valeur d'un élément à gauche est remplacée par la valeur d'un élément à droite. Si un élément à droite est un paramètre, il provient de la primitive montrée comme un événement d'entrée.
xxx	Un nom de paramètre. Exemple: Identifier:= reason signifie que la valeur d'un paramètre 'reason' est attribuée à un paramètre appelé 'Identifier.'
"xxx"	Indique une valeur fixe. Exemple: Identifier:= "abc" signifie que la valeur "abc" est attribuée à un paramètre nommé 'Identifier.'
= ou ==	Une condition logique pour indiquer qu'un élément à gauche est égal à un élément à droite.
<	Une condition logique pour indiquer qu'un élément à gauche est inférieur à un élément à droite.
>	Une condition logique pour indiquer qu'un élément à gauche est supérieur à un élément à droite.
<> or !=	Une condition logique pour indiquer qu'un élément à gauche n'est pas égal à un élément à droite.
>>	Une condition sémantique avec la signification «plus récent».
<<	Une condition sémantique avec la signification «plus ancien».
&&	"AND" (c'est-à-dire: ET) logique
	"OR" (c'est-à-dire: OU) logique
<=	Une condition logique pour indiquer qu'un élément à gauche est inférieur ou égal à l'élément à droite.
>=	Une condition logique pour indiquer qu'un élément à gauche est supérieur ou égal à l'élément à droite.

Convention	Signification
for (Identifier:= start_value to end_value) actions endfor	Cette construction permet l'exécution d'une séquence d'actions dans une boucle au sein d'une même transition. La boucle est exécutée pour toutes les valeurs de start_value à end_value.
If (condition) actions else actions	Cette construction permet l'exécution d'actions alternatives dépendant d'une certaine condition (qui pourrait être la valeur d'un certain identificateur ou le résultat d'une action précédente) au sein d'une même transition. Les parties commençant par "else" peuvent être omises s'il n'y a aucune action si la condition n'est pas remplie.

Les conventions utilisées pour les services dans les diagrammes d'états sont montrées dans le Tableau 4.

Tableau 4 – Conventions utilisées pour les services dans les diagrammes d'états

Convention	Signification
xxx.req ()	demande selon l'ISO/CEI 7498-1 invoque le service et passe tout paramètre requis les paramètres sont omis et énumérés au sein de la description des diagrammes d'états
xxx.ind ()	indication selon l'ISO/CEI 7498-1 indique l'activation d'un service demandé
xxx.rsp (+), xxx.cnf (+)	réponse selon l'ISO/CEI 7498-1 peut acquitter positivement ou parachever une action antérieurement invoquée par une primitive "request" confirmation selon l'ISO/CEI 7498-1 primitive retournée au demandeur pour acquitter positivement ou parachever une action antérieurement invoquée par une primitive "request" les paramètres sont omis et énumérés au sein de la description des diagrammes d'états
xxx.rsp (-), xxx.cnf (-)	réponse selon l'ISO/CEI 7498-1 peut acquitter négativement ou parachever une action antérieurement invoquée par une primitive "request" confirmation selon l'ISO/CEI 7498-1 primitive retournée au demandeur pour acquitter négativement ou parachever une action antérieurement invoquée par une primitive "request" les paramètres sont omis et énumérés au sein de la description des diagrammes d'états
xxx_req (), xxx_cnf (+), xxx_cnf (-)	cette construction est utilisée pour la description des services locaux et confirmés, qui sont transférés entre des diagrammes d'états les paramètres sont énumérés au sien de la description des diagrammes d'états et peuvent facultativement être placés entre les parenthèses
xxx_ind ()	cette construction est utilisée pour la description de services locaux et indique la réception de ce service les paramètres sont énumérés au sein de la description des diagrammes d'états et peuvent facultativement être placés entre les parenthèses
xxx ()	cette construction est utilisée pour la description des fonctions internes du diagramme d'états (locales) les paramètres sont énumérés au sein de la description des diagrammes d'états et peuvent facultativement être placés entre les parenthèses

Convention	Signification
(), (+), (-)	remplacement de la liste des paramètres associés, décrite dans la définition des primitives avec l'exception que les services et les fonctions locaux peuvent livrer des paramètres entre les parenthèses si le résultat n'est pas nécessaire à d'autres exécutions, les signe + et – peuvent être omis

Il est fortement recommandé aux lecteurs de se référer aux articles concernant les définitions des attributs AREP et CREP, aux fonctions locales et aux définitions des FAL-PDU pour appréhender les diagrammes de protocole. Les lecteurs sont censés avoir une connaissance suffisante de ces définitions et celles-ci sont utilisées sans autres explications.

En outre, les éléments de description suivants sont utilisés:

Caractères génériques dans les noms

name_XXX: "XXX" est utilisé comme chaîne générique pour tous les noms commençant par "name".

L'utilisation typique d'un caractère générique est un Event (événement). Dans ce contexte, il y a autant de transitions d'états qu'il existe d'événements possibles pour ce caractère générique.

Macro conditionnelle

<CONDITIONAL -MACRO-NAME>

<

CONDITION1: macrobody1

CONDITION2: macrobody2

...

>

CONDITION1, CONDITION2, etc. définissent tous les cas possibles pour la macro conditionnelle. L'expression "CONDITIONAL-MACRO-NAME" sert de réceptacle pour le "macrocode" dépendant du résultat de la condition.

Macro de remplacement

<REPLACEMENT-MACRO-NAME>

<

XXX= Name1: macrobody1

XXX= Name2: macrobody2

...

>

Name1, Name2, etc. définissent tous les cas possibles pour la macro de remplacement. L'expression "REPLACEMENT-MACRO-NAME" sert de réceptacle pour le "macrocode" dépendant de la valeur de l'utilisation courante du caractère générique d XXX.

EXEMPLE

<SERVICE_REQ_PARA>

<

XXX=Read: Para1, Para2

XXX=Write: Para1, Para2, Para3

>

Lorsqu'elle est appelée dans

MSAB_XXX.req(<SERVICE_REQ_PARA>)

elle donnera

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

4 Concepts

NOTE La CEI 61158-1, Article 9, décrit les concepts de la description des services de couche d'application et les modèles utilisés dans le présent document.

5 ASE Data type

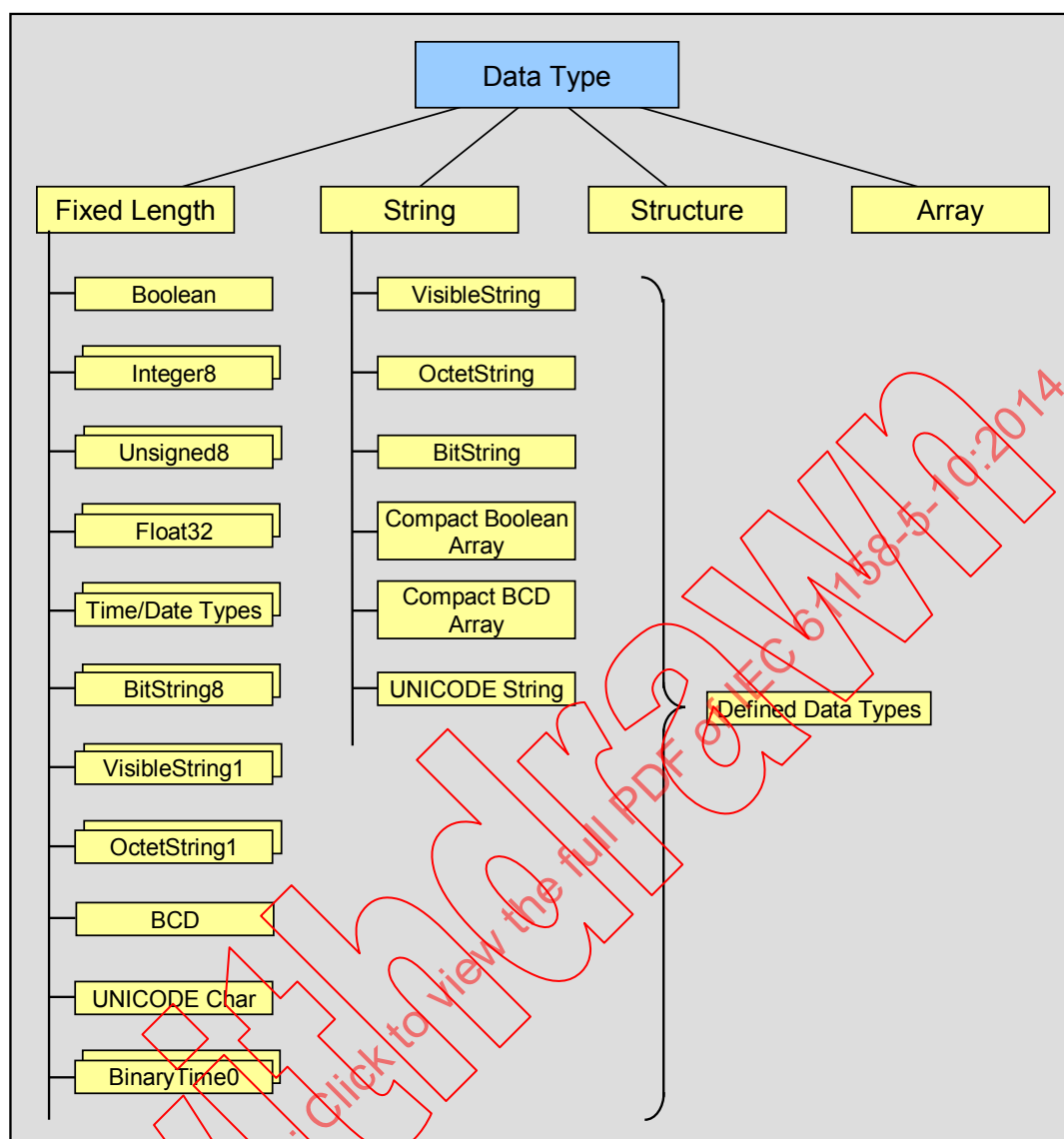
5.1 Généralités

5.1.1 Vue d'ensemble

Les types de données de réseau de terrain spécifient la syntaxe indépendante de tout diagramme pour les données d'application acheminées par les services FAL. La couche d'application de bus de terrain prend en charge la définition et le transfert des types de données tant de base que construits. Les règles de codage pour les types de données spécifiés dans le présent Article sont données dans la CEI 61158-6-10.

Les types de base sont des types atomiques qui ne peuvent pas être décomposés en plus de types élémentaires. Les types construits sont des types composés de types de base et d'autres types construits. Leur complexité et leur profondeur d'imbrication ne sont pas contraintes par la présente norme.

Les types de données sont définis comme des instances de la classe de types de données, comme montré à la Figure 1. Seul un sous-ensemble des types de données définis dans le présent Article est montré dans cette figure. La définition de nouveaux types est accomplie en fournissant un id numérique et en fournissant des valeurs pour les attributs définis pour la classe de types de données.



Légende

Anglais	Français
Defined Data Types	Types de données définis

Figure 1 – Exemple de hiérarchie de la classe de types de données "Data type"

Les définitions des types de données montrées à la Figure 1 sont représentées sous la forme d'une structure classe/format/instance commençant par la classe de types de données intitulée "Data type". Les formats pour les types de données sont définis par la classe de types de données et sont représentés à la Figure 1.

Les classes de données de base sont utilisées pour définir des types de données de longueur fixe et bitstring (chaîne de bits). Les types normalisés issus de l'ISO/CEI 8824-1 sont appelés types de données *simple* (simples). D'autres types de données de base normalisés sont définis spécifiquement pour les applications de bus de terrain et sont appelés types *specific* (spécifiques).

Les types construits spécifiés dans la présente norme sont des chaînes, des matrices et des structures. Il n'y a pas de types normalisés définis pour les matrices et les structures.

5.1.2 Vue d'ensemble de types de base

La plupart des types de base sont définis à partir d'un ensemble de types de l'ISO/CEI 8824-1 (types simples). Certains types de l'ISO/CEI 8824-1 ont été étendus pour une utilisation spécifique dans le bus de terrain (types spécifiques).

Les types simples sont des types universels de l'ISO/CEI 8824-1. Ils sont définis dans la présente norme pour leur fournir des identificateurs de classe de réseau de terrain.

Les types spécifiques sont des types de base définis spécifiquement pour être utilisés dans l'environnement de réseau de terrain. Ils sont définis comme des sous-types de classe simples.

Les types de base ont une longueur constante. Deux variantes sont définies, l'une pour définir les types de données dont la longueur est un nombre entier d'octets et l'autre pour définir les types de données dont la longueur est exprimée en bits.

NOTE Les types Boolean, Integer, OctetString, VisibleString, et UniversalTime sont définis dans la présente norme dans le but de leur attribuer des identificateurs de classes de bus de terrain. La présente norme ne change pas leurs définitions telles que spécifiées dans l'ISO/CEI 8824-1.

5.1.3 Vue d'ensemble des types de longueur fixe

La longueur des types Fixed length (longueur fixe) est un nombre entier d'octets.

5.1.4 Vue d'ensemble des types construits

5.1.4.1 Chaînes

Une chaîne ("string") est composée d'un ensemble ordonné d'un nombre variable d'éléments de longueur fixe typés de façon homogène.

5.1.4.2 Matrices

Une matrice est composée d'un jeu ordonné d'éléments typés de façon homogène. La présente norme n'impose pas de restrictions sur le type de données des éléments de matrice, mais exige par contre que chaque élément soit du même type. Une fois qu'il est défini, le nombre d'éléments dans une matrice ne peut pas être modifié.

5.1.4.3 Structures

Une structure est constituée d'un ensemble ordonné d'éléments typés de façon hétérogène appelés champs. Tout comme dans le cas des matrices, la présente norme ne limite pas le type de données des champs. Cependant, les champs dans une structure peuvent ne pas être du même type.

5.1.4.4 Niveau d'imbrication

La présente norme permet que les matrices et structures contiennent des matrices et des structures. Elle n'impose aucune restriction au nombre de niveaux d'imbrication autorisés. Cependant, les services de la FAL définis pour accéder à des données prévoient l'accès partiel au niveau négocié par le service "initiate". Le nombre par défaut de niveaux pour l'accès partiel est 1 (un).

Lorsqu'une matrice ou une structure contient des éléments construits, l'accès à un élément pris séparément dans sa totalité est toujours assuré. L'accès à des sous-éléments de l'élément construit est également assuré, mais seulement lorsqu'il est négocié de façon explicite au cours de l'établissement de l'AR ou lorsqu'il est préconfiguré de façon explicite sur des AR préétablies.

NOTE Boolean, Integer, OctetString, VisibleString, et UniversalTime sont définis dans la présente norme dans le but de leur attribuer des identificateurs de classes de bus de terrain. Pour accéder à la structure « employé », la FAL permet l'accès indépendant à la structure entière et au champ de premier niveau « nom d'employé. » Sans négocier de manière explicite l'accès partiel à plus d'un niveau, l'accès indépendant à « nom de famille » ou à « prénom » ne serait pas possible; leurs valeurs pourraient être seulement accessibles collectivement comme un bloc par le biais de l'accès à "employé" ou à « nom d'employé ».

5.1.5 Présentation du type de date et d'heure

Les types de données relatifs à la date et à l'heure (à savoir BinaryDate, TimeOfDay avec ou sans indication de date, TimeDifference avec ou sans indication de date, NetworkTime, NetworkTimeDifference et TimeStamp) sont toujours associés à une échelle de temps spécifique.

Les échelles de temps possibles sont les suivantes:

- Heure locale/compteur local; pour exprimer par exemple la durée de travail, etc.
- Horloge de travail; pour exprimer par exemple SendClock, Tinput, Toutput, SampleTime, etc.
- Heure universelle/horloge murale; pour exprimer par exemple un moment comparable, une heure de début, une heure de fin, etc.

Les échelles de temps utilisables ont besoin d'avoir une croissance monotone stricte, comme le TAI (Temps Atomique International), pour permettre les calculs TimeDifference.

Les usages de ces types de données n'ont un sens que s'il existe des informations, à l'échelle de temps desquelles sera associée la valeur.

Seul le type de données TimeStamp indique l'échelle de temps associée et si l'heure est synchronisée. Tous les autres types de données ne donnent pas d'indication de l'échelle de temps à laquelle ils appartiennent ou si l'heure est synchronisée avec d'autres appareils.

NOTE L'IEEE 802.1AS fournit un mécanisme pour synchroniser les appareils par rapport à l'échelle de temps TAI. Ainsi, aucune information UTC (Universal Time Coordinated) n'est disponible.

5.1.6 Spécification des types de données définis par l'utilisateur

Les utilisateurs peuvent trouver nécessaire de définir des types de données personnalisés pour leurs propres applications. Les types définis par l'utilisateur sont pris en charge par la présente norme comme des instances des classes de types de données.

Les types définis par l'utilisateur sont spécifiés de la même manière dont tous les objets de FAL sont spécifiés. Ils sont définis en donnant des valeurs aux attributs spécifiés pour leur classe.

5.1.7 Transfert de données d'utilisateur

Les données utilisateur sont transférées entre applications par le protocole FAL. Tout le codage et le décodage sont accomplis par l'utilisateur de la FAL.

Les règles pour coder les données d'utilisateur dans les unités de données du protocole FAL dépendent du type de données. Ces règles sont définies dans la CEI 61158-6-10. Les types de données définis par l'utilisateur pour lesquels il n'y a pas de règles de codage sont transférés sous la forme d'une séquence de longueur variable d'octets. Le format des données dans l'OctetString est défini par l'utilisateur.

5.1.8 Vue d'ensemble des types de données

Le Tableau 5 donne une vue d'ensemble de tous les types de données définis dans le présent document.

Tableau 5 – Vue d'ensemble des types de données

Data type numeric identifier (Identificateur numérique de type de données)	Data type name	Usage	Pseudonyme	Définition
1	Boolean	la présente norme, CEI 61158-x-3	—	voir 5.3.1.1
2	Integer8	la présente norme, CEI 61158-x-3	char	voir 5.3.1.5.3.1
3	Integer16	la présente norme, CEI 61158-x-3	short	voir 5.3.1.5.3.2
4	Integer32	la présente norme, CEI 61158-x-3	long	voir 5.3.1.5.3.3
5	Unsigned8	la présente norme, CEI 61158-x-3	unsigned char	voir 5.3.1.5.4.1
6	Unsigned16	la présente norme, CEI 61158-x-3	unsigned short	voir 5.3.1.5.4.2
7	Unsigned32	la présente norme, CEI 61158-x-3	unsigned long	voir 5.3.1.5.4.3
8	Float32	la présente norme, CEI 61158-x-3	float	voir 5.3.1.5.1.1
9	VisibleString	la présente norme, CEI 61158-x-3	—	voir 5.3.2.2
10	OctetString	la présente norme, CEI 61158-x-3	—	voir 5.3.2.1
12	TimeOfDay with date indication (avec indication de date)	la présente norme, CEI 61158-x-3	TimeOfDay	voir 5.3.1.4.5
13	TimeDifference	désuet	—	voir 5.3.1.4.4
15	Float64	périphérie décentralisée, automation distribuée	double	voir 5.3.1.5.1.2
39	UNICODEString	la présente norme, CEI 61158-x-3	—	voir 5.3.2.3
50	BinaryDate	la présente norme, CEI 61158-x-3	—	voir 5.3.1.4.1
52	TimeOfDay without date indication (sans indication de date)	la présente norme, CEI 61158-x-3	—	voir 5.3.1.4.3
53	TimeDifference with date indication (avec indication de date)	la présente norme, CEI 61158-x-3	—	voir 5.3.1.4.5
54	TimeDifference without date indication (sans indication de date)	la présente norme, CEI 61158-x-3	—	voir 5.3.1.4.6

Data type numeric identifier (Identificateur numérique de type de données)	Data type name	Usage	Pseudonyme	Définition
55	Integer64	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.3.4
56	Unsigned64	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.4.4
58	NetworkTime	la présente norme, CEI 61158-x-3	—	voir 5.3.1.7.2
59	NetworkTimeDifference	la présente norme, CEI 61158-x-3	—	voir 5.3.1.7.3
60	TimeStamp	la présente norme, CEI 61158-x-3	—	voir 5.3.1.7.4
101	Float32+Unsigned8	la présente norme, CEI 61158-x-3	—	voir 5.3.3.2
102	Unsigned8+Unsigned8	la présente norme, CEI 61158-x-3	—	voir 5.3.3.3
103	OctetString2+Unsigned8	la présente norme, CEI 61158-x-3	—	voir 5.3.3.1
104	Unsigned16_S	la présente norme, CEI 61158-x-3	—	voir 5.3.3.4
105	Integer16_S	la présente norme, CEI 61158-x-3	—	voir 5.3.3.5
106	Unsigned8_S	la présente norme, CEI 61158-x-3	—	voir 5.3.3.6
107	OctetString_S	la présente norme, CEI 61158-x-3	—	voir 5.3.3.7
110	F message trailer with 4 octets («Queue de message F avec quatre octets»)	la présente norme, CEI 61158-x-3	—	voir 5.3.3.8
111	F message trailer with 5 octets («Queue de message F avec cinq octets»)	la présente norme, CEI 61158-x-3	—	voir 5.3.3.9
113	N2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.3.5
114	N4	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.3.6
115	V2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.2.1
116	L2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.2.2

Data type numeric identifier (Identificateur numérique de type de données)	Data type name	Usage	Pseudonyme	Définition
117	R2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.8
118	T2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.5
119	T4	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.6
120	D2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.7
121	E2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.2.1
122	C4	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.3.9
123	X2	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.3.7
124	X4	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.3.8
125	Unipolar2.16	la présente norme, CEI 61158-x-3	—	voir 5.3.1.5.2.2
128 – 255	pour la présente norme, réservé pour la CEI 61158-x-3: types spécifiques à l'utilisateur	CEI 61158-x-3	—	—
256 – 1 024	Réservé	la présente norme	—	voir 5.3.1.6.1
1025	UUID	la présente norme	—	voir 5.3.1.6.1
1 026 – 65 535	Réservé	la présente norme	—	voir 5.3.1.6.1
autre	Réservé	la présente norme, CEI 61158-x-3	—	—

NOTE La lettre x dans CEI 61158-x-3 désigne les parties 3, 4, 5 et 6 spécifiques au type 3 dans la série CEI 61158.

5.2 Définition formelle des objets de types de données

5.2.1 Classe de types de données

5.2.1.1 Modèle

La classe de types de données spécifie la racine de l'arbre de la classe des types de données. Sa classe parent "top" indique le sommet de l'arbre de la classe de FAL.

FAL ASE:	ASE DATA TYPE
CLASS:	DATA TYPE
CLASS ID:	5 (FIXED LENGTH & STRING), 6 (STRUCTURE), 12 (ARRAY)
PARENT CLASS:	TOP

ATTRIBUTS:

1	(m) Attribut-clé:	Data type Numeric Identifier
2	(o) Attribut clé:	Data type Name
3	(m) Attribut:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c) Contrainte:	Format = FIXED LENGTH STRING
4.1	(m) Attribut:	Octet Length
5	(c) Contrainte:	Format = STRUCTURE
5.1	(m) Attribut:	Number of Fields
5.2	(m) Attribut:	List of Fields
5.2.1	(o) Attribut:	Field Name
5.2.2	(m) Attribut:	Field Data type (type de donnée de champ)
6	(c) Contrainte:	Format = ARRAY
6.1	(m) Attribut:	Number of Array Elements
6.2	(m) Attribut:	Array Element Data type (type de donnée d'élément de matrice)

5.2.1.2 Attributs**Data type Numeric Identifier**

Cet attribut identifie l'identificateur numérique du type de données connexe. Le data type numeric identifier (identificateur numérique de type de données) a un type de données Unsigned32.

Data type Name

Cet attribut facultatif identifie le nom du type de données connexe.

Format

Cet attribut identifie le type de données comme étant fixed-length, string, array ou structure de données (data structure).

Octet Length

Cet attribut conditionnel définit la représentation des dimensions de l'objet de type associé. Il est présent lorsque la valeur de l'attribut format est "FIXED LENGTH" ou "STRING". Pour les types de données FIXED LENGTH, il représente la longueur en octets. Pour les types de données STRING, il représente la longueur en octets pour un élément simple d'une chaîne.

Number of Fields

Cet attribut conditionnel définit le nombre de champs dans une structure. Il est présent lorsque la valeur de l'attribut format est "STRUCTURE".

List of Fields

Cet attribut conditionnel est une liste ordonnée de champs contenus dans la structure. Chaque champ est spécifié par son numéro et son type. Les champs sont numérotés séquentiellement à partir de 0 (zéro) dans l'ordre de leur apparition. L'accès partiel à des champs au sein d'une structure est pris en charge en identifiant le champ par son numéro. Cet attribut est présent lorsque la valeur de l'attribut format est "STRUCTURE".

Field Name

Cet attribut facultatif conditionnel spécifie le nom du champ. Il peut être présent lorsque la valeur de l'attribut format est "STRUCTURE".

Field Data type

Cet attribut conditionnel spécifie le type de donnée du champ. Il est présent lorsque la valeur de l'attribut format est "STRUCTURE". Cet attribut peut lui-même spécifier un type de données construit soit en référençant une définition de type de données construit par son id numérique (c'est-à-dire: identificateur numérique), soit en intégrant ici une définition de type

de données construit. Pour intégrer une description, il est utilisé l'attribut "Embedded Data type description" (description intégrée des types de données) qui est montré ci-après.

Number of Array Elements

Cet attribut conditionnel définit le nombre d'éléments de type array (matrice, tableau). Les éléments de matrice sont indicés de "0" à "n-1", la taille de la matrice étant de "n" éléments. Cet attribut est présent lorsque la valeur de l'attribut format est "ARRAY".

Array Element Data type

Cet attribut conditionnel spécifie le type de donnée pour les éléments d'une matrice. Tous les éléments de la matrice ont le même type de donnée. Il est présent lorsque la valeur de l'attribut format est "ARRAY". Cet attribut peut lui-même spécifier un type de données construit soit en référençant une définition de type de données construit par son id numérique (c'est-à-dire: identificateur numérique), soit en intégrant ici une définition de type de données construit. Pour intégrer une description, il est utilisé l'attribut "Embedded Data type description" (description intégrée des types de données) qui est montré ci-après.

Embedded Data type Description

Cet attribut est utilisé pour définir de manière récursive les types de données intégrés et ce, au sein d'une structure ou d'une matrice. Le modèle ci-dessous définit son contenu. Les attributs montrés dans le modèle sont définis ci-dessus dans la classe de types de données, à l'exception de l'attribut "Embedded Data type" (type de données intégré), qui est une référence récursive à cet attribut. Il est utilisé pour définir des éléments imbriqués.

ATTRIBUTES:

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

5.3 Types de données définis pour la FAL

5.3.1 Types de longueur fixe

5.3.1.1 Types Boolean ("booléens")

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

Ce type de données exprime un type de données Boolean avec les valeurs TRUE (non-zéro) et FALSE (zéro).

5.3.1.2 Types Bitstring (chaîne de bits)

5.3.1.2.1 Séquence de bits V2

CLASS: Data type

ATTRIBUTS:

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

Ce type de données exprime un type de données Bitstring consistant en 16 bits simples combinés en deux octets et codés conformément au Tableau 6.

Chaque bit simple a les valeurs TRUE (1) et FALSE (0).

Tableau 6 – Octets V2

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 Quartet L2**CLASS:** Data type**ATTRIBUTS:**

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

Ce type de données exprime un type de données Bitstring consistant en quatre quartets combinés en deux octets. Un quartet (nibble) consiste en quatre bits associés. Le type de données est codé conformément au Tableau 7.

La définition d'un quartet simple n'est pas spécifiée.

Tableau 7 – Octets L2

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

La définition d'un quartet simple n'est pas spécifiée.

5.3.1.3 Types Currency (monnaie)

NOTE Laissé intentionnellement vide

5.3.1.4 Types Date**5.3.1.4.1 BinaryDate****CLASS:** Data type**ATTRIBUTS:**

- 1 Data type Numeric Identifier = 50
- 2 Data type Name = BinaryDate
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 7

Ce type de données est constitué de six éléments de valeurs unsigned (non signées) et exprime l'heure et la date calendaires.

Le premier élément est un type de données Unsigned16 et donne la fraction d'une minute en millisecondes.

Le deuxième élément est un type de données Unsigned8 et donne la fraction d'une heure en minutes.

Le troisième élément est un type de données Unsigned8 et donne la fraction d'un jour en heures, le bit de poids fort indiquant l'heure normalisée (Standard Time) ou l'heure d'économie d'énergie (Daylight Saving Time).

Le quatrième élément est un type de données Unsigned8. Ses trois (3) bits de poids fort donnent le jour de la semaine et ses cinq (5) bits de poids faible donnent le jour du mois.

Le cinquième élément est un type de données Unsigned8 et donne le mois.

Le dernier élément est un type de données Unsigned8 et donne l'année. Les valeurs 0 ... 50 correspondent aux années 2000 à 2050; Les valeurs 51 ... 99 correspondent aux années 1951 à 1999.

Noter que le fuseau horaire est indéterminé.

5.3.1.4.2 TimeOfDay with date indication (avec indication de date)

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

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) et exprime l'heure du jour et la date. Le premier élément est un type de données Unsigned32 et donne l'heure après minuit en millisecondes. Le second élément est un type de données Unsigned16 et donne la date en comptant les jours à partir de 1984-01-01 (1^{er} janvier 1984).

NOTE "TimeOfDay" est un nom d'alias (pseudonyme) pour ce type de données.

5.3.1.4.3 TimeOfDay without date indication (sans indication de date)

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= 52
2	Data type Name	= TimeOfDay without date indication (sans indication de date)
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

Ce type de données est constitué d'un seul élément d'une valeur non signée et exprime l'heure du jour sans indication de date. L'élément est un type de données Unsigned32 et donne l'heure après minuit en millisecondes.

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 ou 6

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) qui expriment la différence de temps. Le premier élément est un type de données Unsigned32 qui donne une partie fractionnaire d'un jour en millisecondes. Le second élément facultatif est un type de données Unsigned16 qui donne la différence en jours.

Ce type de données est désuet; utiliser les types de données "TimeDifference with date indication" (voir 5.3.1.4.2) ou "TimeDifference without date indication" (voir 5.3.1.4.3) à la place.

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

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) qui expriment la différence de temps. Le premier élément est un type de données Unsigned32 qui donne une partie fractionnaire d'un jour en millisecondes. Le second élément est un type de données Unsigned16 qui donne la différence en nombre de jours.

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

Ce type de données est constitué d'un seul élément d'une valeur unsigned (non signée) qui exprime la différence de temps. L'élément est un type de données Unsigned32 qui donne la partie fractionnaire d'un jour en millisecondes.

5.3.1.5 Types numériques

5.3.1.5.1 Types Floating Point (virgule flottante)

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

Ce type a une longueur de quatre octets. Le format de float32 est celui défini par l'IEEE 754 comme étant en simple précision ("single precision").

NOTE "float" est un pseudonyme pour ce type de données.

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

Ce type a une longueur de huit octets. Le format de float64 est celui défini par l'IEEE 754 comme étant en double précision ("double precision").

NOTE "double" est un pseudonyme pour ce type de données.

5.3.1.5.2 Types virgule fixe (Fixed point)

5.3.1.5.2.1 Valeur virgule fixe E2

CLASS: Data type

ATTRIBUTS:

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

Ce type de données est une valeur signée linéaire à virgule fixe qui a une longueur de deux octets ayant un bit de signe, huit binaires avant la virgule décimale et sept binaires après celle-ci. Les octets sont codés conformément au Tableau 8.

Tableau 8 – Octets E2

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
	signe	integer						
Octet 2	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
	entier	fraction						

La plage de valeurs de E2 est montrée dans le Tableau 9.

Tableau 9 – Plage de valeurs de E2

Plage de valeurs	Résolution
$-256+2^{-7} - 256-2^{-7}$	$2^{-7} = 0,0078125$

5.3.1.5.2.2 Unipolar2.16

CLASS: Data type

ATTRIBUTS:

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

Ce type de données est une valeur non signée linéaire à virgule fixe avec deux octets ayant deux binaires avant la virgule décimale et quatorze binaires après celle-ci. Les octets sont codés conformément au Tableau 10 avec une plage de valeurs du Tableau 11. Le type de données exprime une valeur en pourcentage d'un intervalle.

Tableau 10 – Octets Unipolar2.16

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					
Octet 2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}
	fraction							

Tableau 11 – Plage de valeurs de Unipolar2.16

Plage de valeurs	Résolution
$0 \% \leq i \leq (400 \cdot 2^{-14}) \%$	$2^{-14} = 0,0061 \%$

NOTE Ce type est exprimé dans la CEI 60870 comme « nombre virgule fixe non signé » et est également utilisé dans la CEI 61375-1.

5.3.1.5.3 Types Integer (entiers)

5.3.1.5.3.1 Integer8

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 2
2	Data type Name	= Integer8
3	Format	= FIXED LENGTH
4.1	Octet Length	= 1

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à un octet.

NOTE “char” est un pseudonyme pour ce type de données.

5.3.1.5.3.2 Integer16

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 3
2	Data type Name	= Integer16
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à deux octets.

NOTE “short” est un pseudonyme pour ce type de données.

5.3.1.5.3.3 Integer32

CLASS:

Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 4
2	Data type Name	= Integer32
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

Ce type integer est un nombre binaire en complément à deux avec une longueur égale à quatre octets.

NOTE “long” est un pseudonyme pour ce type de données.

5.3.1.5.3.4 Integer64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 55
2	Data type Name	= Integer64
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à huit octets.

5.3.1.5.3.5 Valeur normalisée N2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 113
2	Data type Name	= N2
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

Ce type entier est le même que Integer16, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à deux octets; mais il a toutefois une interprétation spéciale.

La valeur fournit une valeur normalisée linéaire, faisant que la valeur entière 2^{14} (c'est-à-dire 16 384) soit égale à 100 %. La valeur est donc interprétée selon la formule suivante:

$$\text{Valeur normalisée N2 (en \%)} = \text{Valeur} / 16\,384 \quad (1)$$

où

Valeur normalisée N2 est la valeur normalisée linéaire calculée
Valeur est la valeur brute pour l'interprétation

Le type de données offre donc une résolution de 0,006 1 % (= 1 / 16 384). La plage de valeurs de N2 est montrée dans le Tableau 12.

Tableau 12 – Plage de valeurs de N2

Valeur	Signification
-32 768	-200 %
-32 767 ... -1	-199,993 896 484 375 % ... -0,006 1 %
-16 384	-100 %
0	0 %
16 384	100 %
1 ... 32 766	0,006 1 % ... 199,987 792 968 75 %
32 767	199,993 896 484 375 %

5.3.1.5.3.6 Valeur normalisée N4

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 114
2	Data type Name	= N4

3 Format = FIXED LENGTH

4.1 Octet Length = 4

Ce type entier est le même que Integer32, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à quatre octets; mais il a toutefois une interprétation spéciale.

La valeur fournit une valeur normalisée linéaire, faisant que la valeur entière 2^{30} (c'est-à-dire 1 073 741 824) soit égale à 100 %. La valeur est donc interprétée selon la formule suivante:

$$\text{Valeur normalisée N4 (en \%)} = \text{Valeur} / 1\,073\,741\,824 \quad (2)$$

où

Valeur normalisée N4 est la valeur normalisée linéaire calculée

Valeur est la valeur brute pour l'interprétation

Le type de données offre donc une résolution approximativement de 0,000 000 093 1 % (= 1 / 1 073 741 824). La plage de valeurs de N4 est montrée dans le Tableau 13.

Tableau 13 – Plage de valeurs de N4

Valeur	Signification
-2 147 483 648	-200 %
-1 073 741 824	-100 %
-2 147 483 647 ... -1	approx. -199,999 999 906 9 % ... -0,000 000 093 1 %
0	0 %
1 ... 2 147 483 646	approx. 0,000 000 093 1 % ... 199,999 999 813 7 %
1 073 741 824	100 %
2 147 483 647	approx. 199,999 999 906 9 %

5.3.1.5.3.7 Variable normalisée X2

CLASS: Data type

ATTRIBUTS:

1 Data type Numeric Identifier = 123
 2 Data type Name = X2
 3 Format = FIXED LENGTH
 4.1 Octet Length = 2

Ce type entier est le même que Integer16, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à deux octets; mais il a toutefois une interprétation spéciale.

La valeur est interprétée comme étant une valeur normalisée linéaire (une valeur exprimée comme un pourcentage). L'interprétation est similaire à celle du type de données N2, cependant la normalisation à 100 % est variable et déterminée par un paramètre complémentaire "X" extérieur au domaine d'application de ce type de données, faisant que la valeur entière 2^X soit égale à 100 %.

NOTE 1 Si la constante complémentaire spécifiée X est égale à 14, la plage de valeurs de X2 est égale à celle du type de données N2.

La valeur est interprétée selon la formule suivante:

$$\text{Valeur normalisée X2 (en \%)} = \text{Valeur} / 2^X \quad (3)$$

où

Valeur normalisée X2 est la valeur normalisée linéaire calculée
Valeur est la valeur brute pour l'interprétation
X est l'exposant pour les calculs, hors du domaine d'application de la définition de datatype

Le type de données offre donc une résolution de $1 / 2^X$ % ou autrement de 2^{-X} . Les bornes inférieure et supérieure de la valeur sont calculées conformément aux formules suivantes:

$$\text{Borne inférieure X2 (en \%)} = -2^{(15-X)} \quad (4)$$

$$\text{Borne supérieure X2 (en \%)} = 2^{(15-X)} - 2^{-X} \quad (5)$$

où

Borne inférieure X2 est la limite inférieure de la valeur calculée
Borne supérieure X2 est la limite supérieure de la valeur calculée
X est l'exposant pour les calculs, hors du domaine d'application de la définition de datatype

La plage de valeurs de X2 pour des valeurs sélectionnées de "X" est montrée dans le Tableau 14.

Tableau 14 – Plage de valeurs de X2

Paramètre "X"	Valeur	Signification
—	-32 768 ... 32 767	$-2^{(15-X)} \% \leq i \leq (2^{(15-X)} - 2^{-X}) \%$
par exemple X = 10	-32 768 ... 32 767	$-3\,200 \% \leq i \leq -3\,199,90 \%$
par exemple X = 12	-32 768 ... 32 767	$-800 \% \leq i \leq 799,98 \%$
par exemple X = 14	-32 768 ... 32 767	$-200 \% \leq i \leq 199,99 \%$

NOTE 2 La valeur de ce paramètre valeur normalisée renvoie à la constante complémentaire spécifiée X. La constante associée X est exigée afin d'interpréter la valeur interne.

5.3.1.5.3.8 Variable normalisée X4

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 124
- 2 Data type Name = X4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Ce type entier est le même que Integer32, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à quatre octets; mais il a toutefois une interprétation spéciale.

La valeur est interprétée comme étant une valeur normalisée linéaire (une valeur exprimée comme un pourcentage). L'interprétation est similaire à celle du type de données N4, cependant la normalisation à 100 % est variable et déterminée par un paramètre complémentaire "X" extérieur au domaine d'application de ce type de données, faisant que la valeur entière 2^X soit égale à 100 %.

NOTE 1 Si la constante complémentaire spécifiée X est égale à 30, la plage de valeurs de X4 est égale à celle du type de données N4.

La valeur est interprétée selon la formule suivante:

$$\text{Valeur normalisée } X4 \text{ (en \%)} = \text{Valeur} / 2^X \quad (6)$$

où

<i>Valeur normalisée X4</i>	est la valeur normalisée linéaire calculée
<i>Valeur</i>	est la valeur brute pour l'interprétation
<i>X</i>	est l'exposant pour les calculs, hors du domaine d'application de la définition de datatype

Le type de données offre donc une résolution de $1 / 2^X$ % ou autrement de 2^{-X} . Les bornes inférieure et supérieure de la valeur sont calculées conformément aux formules suivantes:

$$\text{Borne inférieure } X4 \text{ (en \%)} = -2^{(31-X)} \quad (7)$$

$$\text{Borne supérieure } X4 \text{ (en \%)} = 2^{(31-X)} - 2^{-X} \quad (8)$$

où

<i>Borne inférieure X4</i>	est la limite inférieure de la valeur calculée
<i>Borne supérieure X4</i>	est la limite supérieure de la valeur calculée
<i>X</i>	est l'exposant pour les calculs, hors du domaine d'application de la définition de datatype

La plage de valeurs de X4 pour des valeurs sélectionnées de "X" est montrée dans le Tableau 15.

Tableau 15 – Plage de valeurs de X4

Paramètre "X"	Valeur	Signification
—	-2 147 483 648 ... 2 147 483 647	$-2^{(31-X)} \% \leq i \leq (2^{(31-X)} - 2^{-X}) \%$
par exemple X = 26	-2 147 483 648 ... 2 147 483 647	$-3\,200 \% \leq i \leq -3\,200 \%$
par exemple X = 28	-2 147 483 648 ... 2 147 483 647	$-800 \% \leq i \leq 800 \%$
par exemple X = 30	-2 147 483 648 ... 2 147 483 647	$-200 \% \leq i \leq 200 \%$

NOTE 2 La valeur de ce paramètre valeur normalisée renvoie à la constante complémentaire spécifiée X. La constante associée X est exigée afin d'interpréter la valeur interne.

5.3.1.5.3.9 Valeur virgule fixe C4

CLASS: Data type

ATTRIBUTS:

1	Data type Numeric Identifier	= 122
2	Data type Name	= C4
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

Ce type de données est le même que Integer32, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à quatre octets; mais il a toutefois une interprétation spéciale.

La valeur est interprétée comme étant une valeur linéaire à virgule fixe avec quatre décimales selon la formule suivante:

$$\text{Valeur à virgule fixe C4} = \text{Valeur} / 10\,000 \quad (9)$$

où

Valeur virgule fixe C4 est la valeur virgule fixe linéaire calculée

Valeur est la valeur brute pour l'interprétation

Le type de données offre donc une résolution de 0,000 1 (= 1 / 10 000). La plage de valeurs de C4 est montrée dans le Tableau 16.

Tableau 16 – Plage de valeurs de C4

Valeur	Signification
-2 147 483 648	-214 748,364 8
0	0
2 147 483 647	214 748,364 7

5.3.1.5.4 Types Unsigned (non signés)

5.3.1.5.4.1 Unsigned8

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 5
- 2 Data type Name = Unsigned8
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 1

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur égale à un octet.

NOTE "unsigned char" est un pseudonyme pour ce type de données.

5.3.1.5.4.2 Unsigned16

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 6
- 2 Data type Name = Unsigned16
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de deux octets.

NOTE "unsigned short" est un pseudonyme pour ce type de données.

5.3.1.5.4.3 Unsigned32

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 7
- 2 Data type Name = Unsigned32
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de quatre octets.

NOTE “unsigned long” est un pseudonyme pour ce type de données.

5.3.1.5.4.4 Unsigned64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 56
2	Data type Name	= Unsigned64
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de huit octets.

5.3.1.5.5 Constante de temps T2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 118
2	Data type Name	= T2
3	Format	= FIXED LENGTH
4.1	Octet Length	= 2

Ce type entier est le même que Unsigned16, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à deux octets; mais il a toutefois une interprétation spéciale.

La valeur fournit une valeur de temps basée sur un temps d'échantillonnage défini “Ta” extérieur au domaine d'application de ce type de données.

La valeur est interprétée selon la formule suivante:

$$\text{Valeur de temps} = \text{Valeur} * T_a \quad (10)$$

où

Valeur de temps est la valeur de temps calculée

Valeur est la valeur brute pour l'interprétation

T_a est le temps d'échantillonnage défini, hors du domaine d'application de ce type de données

Le type de données offre donc une résolution de “Ta”. La plage de valeurs de T2 est montrée dans le Tableau 17. Noter que la plage de valeurs est limitée; les valeurs à l'extérieur du résultat fournissent une valeur de temps de “0”.

Tableau 17 – Plage de valeurs de T2

Valeur	Signification
0 – 32 767	Valeur * Ta
32 768 – 65 535	0

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié Ta. Le temps d'échantillonnage associé Ta est exigé afin d'interpréter la valeur interne.

5.3.1.5.6 Constante de temps T4

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 119

- 2 Data type Name = T4
3 Format = FIXED LENGTH
4.1 Octet Length = 4

Ce type entier est le même que Unsigned32, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à quatre octets; mais il a toutefois une interprétation spéciale.

La valeur fournit une valeur de temps basée sur un temps d'échantillonnage défini "Ta" extérieur au domaine d'application de ce type de données.

La valeur est interprétée selon la formule suivante:

$$\text{Valeur de temps} = \text{Valeur} * T_a \quad (11)$$

où

<i>Valeur de temps</i>	est la valeur de temps calculée
<i>Valeur</i>	est la valeur brute pour l'interprétation
T_a	est le temps d'échantillonnage défini, hors du domaine d'application de ce type de données

Le type de données offre donc une résolution de "Ta". La plage de valeurs de T4 est montrée dans le Tableau 18. Contrairement au cas du type de données T2, la plage de valeurs n'est pas limitée.

Tableau 18 – Plage de valeurs de T2

Valeur	Signification
0 – 4 294 967 295	Valeur * Ta

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié Ta. Le temps d'échantillonnage associé Ta est exigé afin d'interpréter la valeur interne.

5.3.1.5.7 Constante de temps D2

CLASS: Data type

ATTRIBUTS:

- 1 Data type Numeric Identifier = 120
2 Data type Name = D2
3 Format = FIXED LENGTH
4.1 Octet Length = 2

Ce type entier est le même que Unsigned16, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à deux octets; mais il a toutefois une interprétation spéciale.

La valeur fournit une valeur de temps basée sur un temps d'échantillonnage défini "Ta" extérieur au domaine d'application de ce type de données.

La valeur est interprétée selon la formule suivante:

$$\text{Valeur de temps} = \text{Valeur} \times (T_a / 16\,384) \quad (12)$$

où

<i>Valeur de temps</i>	est la valeur de temps calculée
<i>Valeur</i>	est la valeur brute pour l'interprétation
T_a	est le temps d'échantillonnage défini, hors du domaine d'application de ce type de données

Le type de données offre donc une fraction du temps d'échantillonnage constant avec une résolution de " $T_a / 16\ 384$ ". La plage de valeurs de D2 est montrée dans le Tableau 19. Noter que la plage de valeurs est limitée; les valeurs à l'extérieur du résultat fournissent une valeur de temps de "0".

Tableau 19 – Plage de valeurs de D2

Valeur	Signification
0 – 32 767	Valeur * ($T_a / 16\ 384$)
32 768 – 65 535	0

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié T_a . Le temps d'échantillonnage associé T_a est exigé afin d'interpréter la valeur interne.

5.3.1.5.8 Constante de temps R2

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 117
- 2 Data type Name = R2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Ce type entier est le même que Unsigned16, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à deux octets; mais il a toutefois une interprétation spéciale.

La valeur fournit une valeur de temps basée sur un temps d'échantillonnage défini " T_a " extérieur au domaine d'application de ce type de données.

La valeur est interprétée selon la formule suivante:

$$\text{Valeur de temps} = 16\ 384 * (T_a / \text{Valeur}) \quad (13)$$

où

Valeur de temps est la valeur de temps calculée

T_a est le temps d'échantillonnage défini, hors du domaine d'application de définition de type de données

Valeur est la valeur brute pour l'interprétation

Le type de données offre donc un multiple inverse du temps d'échantillonnage constant avec une résolution de " T_a / Valeur ". La plage de valeurs de R2 est montrée dans le Tableau 20. Noter que la plage de valeurs est limitée; les valeurs à l'extérieur du résultat fournissent une valeur de temps de "0".

Tableau 20 – Plage de valeurs de R2

Valeur	Signification
0	16 384
1 – 16 384	16 384 * (Valeur / T_a)
16 385 – 65 535	16 384

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié T_a . Le temps d'échantillonnage associé T_a est exigé afin d'interpréter la valeur interne.

5.3.1.6 Types de caractère OctetString

5.3.1.6.1 UUID

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 1025
- 2 Data type Name = UUID
- 3 Format = STRUCTURE
- 5.1 Number of Fields = 4
- 5.2.1 Field Name = Data1
- 5.2.2 Field Data type = Unsigned32
- 5.2.3 Field Name = Data2
- 5.2.4 Field Data type = Unsigned16
- 5.2.5 Field Name = Data3
- 5.2.6 Field Data type = Unsigned16
- 5.2.7 Field Name = Data4
- 5.2.8.1 Format = ARRAY
- 5.2.8.4.1 Number of Array Elements = 8
- 5.2.8.4.2 Array Element Data type = Unsigned8

Ce type de données définit un type de données de longueur fixe égale à 16 octets. La sémantique est spécifiée par le document The Open Group – Publication C706 et ne relève pas du domaine d'application de la présente norme.

Ce type de données est structuré comme suit.

Data1

Ce champ contient les huit premiers chiffres hexadécimaux de l'identifiant unique universel («Universally Unique Identifier» (UUID)).

Data2

Ce champ contient le premier groupe de quatre chiffres hexadécimaux de l'UUID.

Data3

Ce champ contient le deuxième groupe de quatre chiffres hexadécimaux de l'UUID.

Data4

Ce champ contient une matrice de huit éléments. Les deux premiers éléments spécifient le troisième groupe de quatre chiffres hexadécimaux de l'UUID. Les six éléments restants spécifient les 12 derniers chiffres hexadécimaux de l'UUID.

Les valeurs spécifiques pour les éléments de périphériques décentralisés sont montrées dans le Tableau 21.

Tableau 21 – UUID pour périphériques décentralisés

Valeur	Description
UUID_NIL 00000000-0000-0000-0000-000000000000	AR par défaut dans un appareil
UUID_IO_ObjectInstance_XYZ DEA00000-6C97-11D1-8271-{xxxxxxxxxxxx}	Identifie une instance d'objet spéciale au sein d'un appareil physique au cas où il y en aurait plus d'une, où - xxxx représente le numéro d'instance ou de nœud - yyyy identifie le Device ID (identificateur d'appareil) comme étant un numéro spécifique à chaque fournisseur pour la classe d'appareil et - zzzz représente le Vendor ID (identificateur de vendeur) comme étant un numéro administratif central.
UUID_IO_DeviceInterface DEA00001-6C97-11D1-8271-00A02442DF7D	Identifie de manière univoque l'interface de l'appareil E/S.

Valeur	Description
UUID_IO_ControllerInterface DEA00002-6C97-11D1-8271-00A02442DF7D	Identifie de manière univoque l'interface de l'appareil de commande E/S.
UUID_IO_SupervisorInterface DEA00003-6C97-11D1-8271-00A02442DF7D	Identifie de manière univoque l'interface de l'appareil de surveillance E/S.
UUID_IO_ParameterServerInterface DEA00004-6C97-11D1-8271-00A02442DF7D	Identifie de manière univoque l'interface du serveur de paramètres E/S.

5.3.1.7 Types Time (temps)

5.3.1.7.1 Généralités

La limitation de 5.1.5 s'applique.

5.3.1.7.2 NetworkTime

CLASS: Data type

ATTRIBUTS:

1	Data type Numeric Identifier	= 58
2	Data type Name	= NetworkTime
3	Format	= FIXED LENGTH
4.1	Octet Length	= 8

Ce type de données est le même que le type Unsigned64. Son interprétation est basée sur IETF RFC 5905 et exprime l'heure du réseau relative à une date particulière comme illustré dans la (voir Figure 2).

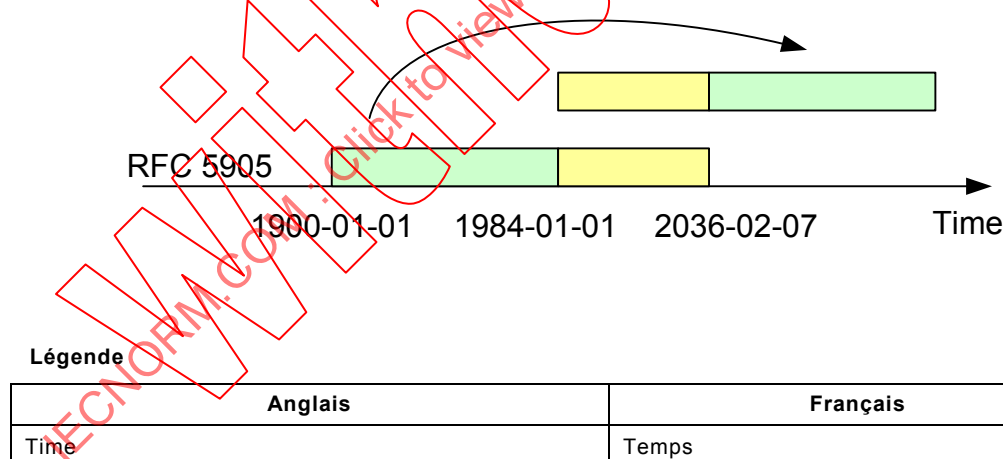


Figure 2 – Relation de date de NetworkTime

La valeur de NetworkTime décrit un temps basé sur le fuseau horaire TUC (Temps Universel Coordonné, aussi appelé Greenwich Mean Time) en unités de $1/2^{32}$ s depuis 1900-01-01 T 00:00:00 Z pour les valeurs de temps supérieures ou égales à 0x9DFF440000000000 ou sinon depuis 2036-02-07 T 06:28:16 Z pour les valeurs de temps inférieures à 0x9DFF440000000000, qui représente 1984-01-01 T 00:00:00 Z.

NOTE L'heure est représentée conformément à l'ISO/CEI 8601.

Les passages par zéro au bout de 136 ans ne sont pas détectables automatiquement et doivent être maintenus par l'application.

Le bit de poids faible de la partie fractionnaire (2^0) est utilisé conformément au Tableau 22.

Tableau 22 – Bit d'état de poids faible de la partie fractionnaire (2⁰)

Valeur	Signification
0	Valeur NetworkTime relative à l'heure synchronisée universelle
1	Valeur NetworkTime relative à l'heure locale

5.3.1.7.3 NetworkTimeDifference**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 59
- 2 Data type Name = NetworkTimeDifference
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 8

Ce type de données est le même que Integer64, c'est-à-dire un nombre binaire en complément à deux avec une longueur égale à huit octets. La valeur est interprétée comme la différence entre deux valeurs NetworkTime sous forme de parties fractionnaires de secondes en $1/2^{32}$ s.

5.3.1.7.4 TimeStamp**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 60
- 2 Data type Name = TimeStamp
- 3 Format = STRUCTURE
- 3.1 Number of Fields = 4
- 3.2.1 Field Name = Status
- 3.2.2 Field Data type = Unsigned16
- 3.2.3 Field Name = SecondsHigh
- 3.2.4 Field Data type = Unsigned16
- 3.2.5 Field Name = SecondsLow
- 3.2.6 Field Data type = Unsigned32
- 3.2.7 Field Name = Nanoseconds
- 3.2.8 Field Data type = Unsigned32

Ce type de données définit un type de données de longueur fixe égale à 12 octets.

La valeur est interprétée selon la formule suivante:

$$\text{TimeStamp.Seconds} = (\text{SecondsHigh} \ll 32) + \text{SecondsLow} \quad (14)$$

$$\text{TimeStamp.Nanoseconds} = \text{Nanoseconds}$$

où

- TimeStamp.Seconds* est la partie "secondes" de la valeur calculée du marqueur temporel (une valeur de 48 bits)
- TimeStamp.Nanoseconds* est la partie "nanosecondes" de la valeur calculée du marqueur temporel
- SecondsHigh* est la valeur brute pour l'interprétation
- SecondsLow* est la valeur brute pour l'interprétation
- Nanoseconds* est la valeur brute pour l'interprétation

Un TimeStamp est construit selon le Tableau 23 et les définitions suivantes:

- Si l'appareil est synchronisé et est en synchronisation avec l'heure universelle principale, le TimeStamp est relatif à l'heure universelle
- Autrement, si l'appareil a un moyen pour obtenir une heure locale, le TimeStamp est relatif à l'heure locale.

Tableau 23 – Plage de valeurs d'état

Value	Meaning
0	Valeur TimeStamp relative à l'heure synchronisée universelle
1	Valeur TimeStamp relative à l'heure locale
2	Valeur TimeStamp relative à l'heure locale (échelle de temps arbitraire)
autre	réservé

Si l'heure ne dispose pas d'une mémoire tampon secourue par pile lorsque l'appareil est hors tension, l'état est défini sur une « échelle de temps arbitraire d'heure locale » jusqu'à ce que l'appareil soit synchronisé par rapport à l'heure universelle.

Si la synchronisation par rapport à l'heure universelle est perdue, l'état passe à l'« heure locale » jusqu'à ce que la mise hors tension.

5.3.1.8 Types de caractère VisibleString

NOTE laissé intentionnellement vide

5.3.2 Types String (chaîne)

5.3.2.1 OctetString

CLASS: Data type

ATTRIBUTS:

1	Data type Numeric Identifier	= 10
2	Data type Name	= OctetString
3	Format	= STRING
4.1	Octet Length	= 1 à n

Un OctetString est une séquence ordonnée d'octets, numérotés de 1 à n. Pour les besoins du débat, l'octet 1 de la séquence est appelé le premier octet. La CEI 61158-6-10 définissent l'ordre d'émission.

NOTE La CEI 61158-6-3 applique la définition de la CEI 61158-6-10.

5.3.2.2 VisibleString

CLASS: Data type

ATTRIBUTS:

1	Data type Numeric Identifier	= 9
2	Data type Name	= VisibleString
3	Format	= STRING
4.1	Octet Length	= 1 à n

Ce type est défini comme étant le type "string" de la version de référence internationale ISO/CEI 646 (International Reference Version) sans le caractère "del" (codage 0x7F).

5.3.2.3 UNICODEString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 39
2	Data type Name	= UNICODEString
3	Format	= STRING
4.1	Octet Length	= 1 à n

Ce type est défini comme étant le type "string" (chaîne) de l'ISO/CEI 10646 avec toutes les variantes de codage ex: UTF-8, UTF-16 et UTF-32.

5.3.3 Types Structure

5.3.3.1 OctetString2+Unsigned8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 103
2	Data type Name	= OctetString2+Unsigned8
3	Format	= STRUCTURE
5.1	Number of Fields	= 2
5.2	List of Fields	
5.2.1	Field Name	= Value
5.2.2	Field Data type	= OctetString[2]
5.2.3	Field Name	= Status
5.2.4	Field Data type	= Unsigned8

Ce type de données définit le type de données OctetString2+Unsigned8 et est codé conformément au Tableau 24.

Tableau 24 – OctetString2+Unsigned8

Bits	7	6	5	4	3	2	1	0
Octet 1	1. octet							
Octet 2	2. octet							
Octet 3	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰

Ce type de données est structuré comme suit.

Value

Ce champ contient la valeur comme OctetString avec la longueur de 2.

Status

Ce champ décrit le statut de la valeur comme qualificateur.

5.3.3.2 Float32+Unsigned8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 101
2	Data type Name	= Float32+Unsigned8
3	Format	= STRUCTURE
5.1	Number of Fields	= 2
5.2	List of Fields	
5.2.1	Field Name	= Value
5.2.2	Field Data type	= Float32
5.2.3	Field Name	= Status
5.2.4	Field Data type	= Unsigned8

Ce type de données définit le type de données Float32+Unsigned8 et est codé conformément au Tableau 25.

SN: signe 0 = positif, 1 = négatif

Tableau 25 – Octets de Float32+Unsigned8

Bits	7	6	5	4	3	2	1	0
Octet 1	Exposant (E)							
	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Octet 2	(E)	Fraction (F)						
	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Octet 3	Fraction (F)							
	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}
Octet 4	Fraction (F)							
	2^{-16}	2^{-17}	2^{-18}	2^{-19}	2^{-20}	2^{-21}	2^{-22}	2^{-23}
Octet 5	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Ce type de données est structuré comme suit.

Valeur

Ce champ contient la valeur comme Float32.

Status

Ce champ décrit le statut de la valeur comme qualificateur.

5.3.3.3 Unsigned8+Unsigned8

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 102
- 2 Data type Name = Unsigned8+Unsigned8
- 3 Format = STRUCTURE
- 5.1 Number of Fields = 2
- 5.2 List of Fields
 - 5.2.1 Field Name = Value
 - 5.2.2 Field Data type = Unsigned8
 - 5.2.3 Field Name = Status
 - 5.2.4 Field Data type = Unsigned8

Ce type de données définit le type de données Unsigned8+Unsigned8 et est codé conformément au Tableau 26.

Tableau 26 – Octets de Unsigned8+Unsigned8

Bits	7	6	5	4	3	2	1	0
Octet 1	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Octet 2	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Ce type de données est structuré comme suit.

Valeur

Ce champ contient la valeur comme Unsigned8.

Status

Ce champ décrit le statut de la valeur comme qualificateur.

5.3.3.4 Unsigned16_S

CLASS: Data type

ATTRIBUTS:

- 1 Data type Numeric Identifier = 104
- 2 Data type Name = Unsigned16_S
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Ce type de données est structuré et codé conformément au Tableau 27 et au Tableau 28.

Tableau 27 – Octets Unsigned16_S

Bits	7	6	5	4	3	2	1	0
Octets 1	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6
Octets 2	2^5	2^4	2^3	2^2	2^1	2^0	St1	St0

Tableau 28 – Signification de Unsigned16_S

St1 (bit1)	St0 (bit0)	Signification
0	0	voie d'entrée: mauvaise (la valeur est une valeur de sécurité intégrée) voie de sortie: reserved (réserve)
0	1	voie d'entrée: simulation voie de sortie: reserved (réserve)
1	0	voie d'entrée: incertaine voie de sortie: reserved (réserve)
1	1	voie d'entrée: mauvaise voie de sortie: reserved (réserve)

5.3.3.5 Integer16_S

CLASS: Data type

ATTRIBUTS:

- 1 Data type Numeric Identifier = 105
- 2 Data type Name = Integer16_S
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Ce type de données est structuré et codé conformément au Tableau 29 et au Tableau 30.

SN: signe 0 = positif, 1 = négatif

Tableau 29 – Octets de Integer16_S

Bits	7	6	5	4	3	2	1	0
Octets 1	SN	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6
Octets 2	2^5	2^4	2^3	2^2	2^1	2^0	St1	St0

Tableau 30 – Signification de Integer16_S

St1 (bit1)	St0 (bit0)	Signification
0	0	voie d'entrée: mauvaise (la valeur est une valeur de sécurité intégrée) voie de sortie: reserved (réserve)
0	1	voie d'entrée: simulation voie de sortie: reserved (réserve)

St1 (bit1)	St0 (bit0)	Signification
1	0	voie d'entrée: incertaine voie de sortie: reserved (réservé)
1	1	voie d'entrée: mauvaise voie de sortie: reserved (réservé)

5.3.3.6 Unsigned8_S

CLASS: Data type

ATTRIBUTS:

1	Data type Numeric Identifier	= 106
2	Data type Name	= Unsigned8_S
3	Format	= FIXED LENGTH
4.1	Octet Length	= 1

Ce type de données est structuré et codé conformément au Tableau 31 et au Tableau 32.

Tableau 31 – Octets de Unsigned8_S

Bits	7	6	5	4	3	2	1	0
Octets 1	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰	St1	St0

Tableau 32 – Signification de Unsigned8_S

St1 (bit1)	St0 (bit0)	Signification
0	0	voie d'entrée: mauvaise (la valeur est une valeur de sécurité intégrée) voie de sortie: reserved (réservé)
0	1	voie d'entrée: simulation voie de sortie: reserved (réservé)
1	0	voie d'entrée: incertaine voie de sortie: reserved (réservé)
1	1	voie d'entrée: mauvaise voie de sortie: reserved (réservé)

5.3.3.7 OctetString_S

CLASS: Data type

ATTRIBUTS:

1	Data type Numeric Identifier	= 107
2	Data type Name	= OctetString_S
3	Format	= STRING
4.1	Octet Length	= N

Ce type de données est structuré et codé conformément au Tableau 33 et au Tableau 34.

Tableau 33 – Octets de OctetString_S

Bits	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Octet 1	ch(8)	ch(7)	ch(6)	ch(5)	ch(4)	ch(3)	ch(2)	ch(1)
***	***	***	***	***	***	***	***	***
Octet m	ch(n)	ch(n-1)	ch(n-2)	ch(n-3)	ch(n-4)	ch(n-5)	ch(n-6)	ch(n-7)
Octet m+1	st1(4)	st0(4)	st1(3)	st0(3)	st1(2)	st0(2)	st1(1)	st0(1)

5.3.3.9 F message trailer with 5 octets («Queue de message F avec cinq octets»)

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	= 111
2	Data type Name	= F message trailer with 5 octets (queue de message F avec cinq octets)
3	Format	= FIXED LENGTH
4.1	Octet Length	= 5

Cette structure de données est constituée de l'octet status/control (statut/commande) et des paramètres CRC de 4 octets. Ce type de données peut être associé à des données d'entrée ou de sortie jusqu'à 122 octets.

Ce type de données est structuré et codé conformément au Tableau 36.

Tableau 36 – F message trailer with 5 octets («Queue de message F avec cinq octets»)

Bits	7	6	5	4	3	2	1	0
Octet 1	Octet Status/Control							
Octet 2	1. octet CRC (octet de poids fort)							
Octet 3	2. octet CRC							
Octet 4	3. octet CRC							
Octet 5	4. octet CRC (octet de poids faible)							

5.4 Spécification de service de l'ASE Data type

Il n'y a pas de services opérationnels définis pour l'objet de type.

6 Modèle de communication pour services communs

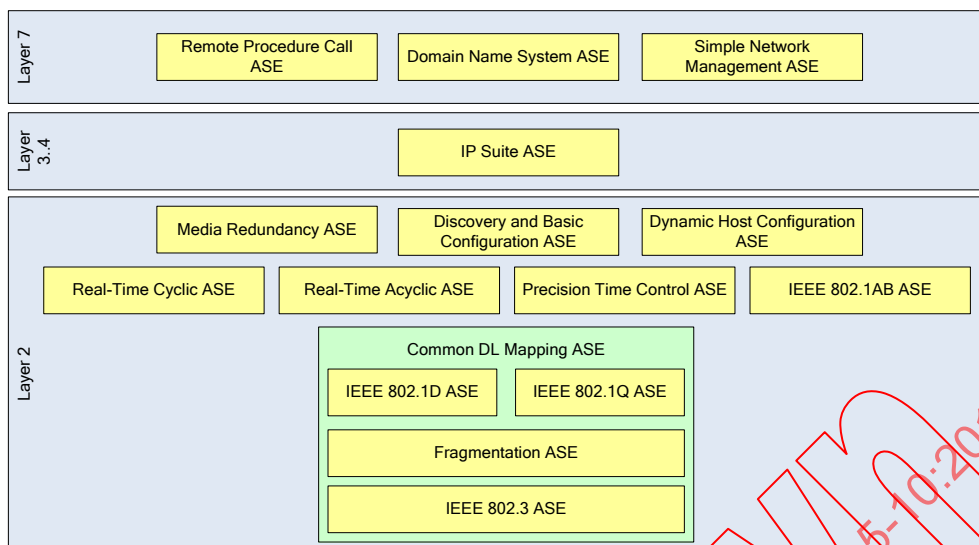
6.1 Concepts

6.1.1 Vue d'ensemble

Les concepts des éléments communs du rapport technique CEI 61158-1 sont appliqués. La présente partie de la spécification définit des services communs, qui sont utilisés pour la périphérie décentralisée. En outre, la présente partie de la spécification définit l'utilisation de raffinements ou d'extensions apporté(e)s aux normes référencées.

6.1.2 ASE de la FAL

La Figure 3 montre les ASE de la FAL des services communs selon leur couche OSI.



Légende

Anglais	Français
Remote Procedure Call ASE	ASE "Appel de procédure distante"
Domain Name System ASE	ASE "Système de noms de domaine")
Simple Network Management ASE	ASE "Gestion de réseau simple"
Layer 7	Couche 7
IP Suite ASE	ASE "Suite IP"
Layer 3.4	Couche 3.4
Media Redundancy ASE	ASE "Redondance de supports"
Discovery and Basic Configuration ASE	ASE "Découverte et configuration de base"
Dynamic Host Configuration ASE	ASE "Configuration d'hôte dynamique"
Real Time Cyclic ASE	ASE "Cyclique temps réel"
Real Time Acyclic ASE	ASE "Acyclique temps réel"
Precision Time Control ASE	ASE "Commande temps de precision"
IEEE 802.1AB ASE	ASE „IEEE 802.1AB“
Common DL Mapping ASE	ASE "Mapping de DL commun"
IEEE 802.1D ASE	ASE "IEEE 802.1D"
IEEE 802.1Q ASE	ASE "IEEE 802.1Q"
Fragmentation ASE	ASE "Fragmentation"
IEEE 802.3 ASE	ASE "IEEE 802.3"
Layer 2	Couche 2

Figure 3 – Architecture de communication des ASE de FAL

6.2 Types de données ASE

Les types de données pris en charge par des services communs sont un sous-ensemble de ceux qui sont définis à l'Article 5. Ce sont:

- Boolean
- BinaryDate
- TimeOfDay with date indication (avec indication de date)
- TimeOfDay without date indication (sans indication de date)
- TimeDifference with date indication (avec indication de date)

- TimeDifference without date indication (sans indication de date)
- Float32
- Float64
- Integer8
- Integer16
- Integer32
- Integer64
- Unsigned8
- Unsigned16
- Unsigned32
- Unsigned64
- UUID
- NetworkTime
- NetworkTimeDifference
- OctetString
- VisibleString

6.3 Éléments de service application

6.3.1 ASE de découverte et de configuration de base

6.3.1.1 Vue d'ensemble

Le concept de découverte et configuration de base est un concept consistant à lire et écrire des paramètres de configuration réseau de base et à découvrir des appareils au moyen de différents critères de filtrage. Il utilise les services de l'IEEE 802.3.

L'ASE « Discovery and basic Configuration Protocol » (Protocole de découverte et de configuration de base (DCP)) est axé sur une interface Ethernet particulière. Bien garder à l'esprit qu'un appareil particulier peut avoir plus d'une interface Ethernet. L'ASE DCP lui-même ne sait pas si, oui ou non, plusieurs interfaces appartiennent au même appareil; l'autre ASE (par exemple, l'ASE de gestion réseau simple ou autre ASE de gestion d'appareil physique) offre ces moyens.

6.3.1.2 Spécification de la classe DCP

6.3.1.2.1 Généralités

L'ASE "DCP" définit un type d'objet DCP.

Certains des attributs de l'ASE DCP peuvent être stockés de façon temporaire ou permanente telle qu'indiquée par le paramètre de service Data Qualifier. Si le Data Qualifier est mis à "temporary" (« temporaire »), l'attribut est réinitialisé à sa valeur par défaut après un cycle d'alimentation. Si le Data Qualifier est mis à "permanent", l'attribut contient sa valeur pendant un cycle d'alimentation. Ainsi, le Data Qualifier détermine la durée de vie de la valeur des attributs. Si l'attribut est positionné, son contenu est remplacé par la nouvelle valeur indépendamment du Data Qualifier. En particulier, il y a exactement un attribut, qui est remplacé indépendamment de la valeur des Data Qualifiers; par exemple, un positionnement sur "temporary" remplace donc un précédent positionnement sur "permanent".

L'ASE DCP contient des éléments qui peuvent être écrits par le service Set. Chacun des éléments inscriptibles dispose d'un attribut supplémentaire qui a l'une des valeurs suivantes:

YES

Le service d'écriture sur cet élément est toujours accepté. Ainsi, la couche de protocole gère le service d'écriture indépendamment de l'application.

NO

Le service d'écriture sur cet élément est toujours refusé. Ainsi, la couche de protocole gère le service d'écriture indépendamment de l'application.

QUERY

Le service d'écriture sur cet élément nécessite l'approbation par l'application. Ainsi la couche de protocole et la couche application gèrent le service d'écriture. Dans ce cas, le service Local Set Command est appelé par la couche du protocole.

6.3.1.2.2 Modèle

Un objet DCP est décrit par le modèle suivant:

ASE: ASE DCP
CLASS: DCP
CLASS ID: pas utilisé
PARENT CLASS: IEEE 802.1AB
ATTRIBUTES:

1	(m) Attribut-clé:	Implicit
2	(m) Attribut:	List of Device Options
2.1	(m) Attribut:	Device Option
2.2	(m) Attribut:	Device Suboption
3	(m) Attribut:	IP Option
3.1	(m) Attribut:	Adresse MAC
3.2	(m) Attribut:	IP Parameter
3.2.1	(m) Attribut:	IP Address
3.2.2	(m) Attribut:	Subnet Mask
3.2.3	(m) Attribut:	Standard Gateway
3.2.4	(m) Attribut:	DNS Server Addresses
4	(m) Attribut:	Device Properties Option
4.1	(m) Attribut:	Name Of Station
4.2	(m) Attribut:	List of Ports
4.2.1	(m) Attribut:	Alias Name Of Station
4.3	(m) Attribut:	Device Vendor
4.4	(m) Attribut:	Device ID
4.5	(m) Attribut:	Device Role
4.6	(m) Attribut:	Device Instance
5	(o) Attribut:	DHCP Option
5.1	(m) Attribut:	Host Name
5.2	(m) Attribut:	Vendor Specific Information
5.3	(m) Attribut:	Server Identifier
5.4	(m) Attribut:	Parameter Request List
5.5	(m) Attribut:	Class Identifier
5.6	(m) Attribut:	DHCP Client Identifier
5.7	(m) Attribut:	Fully Qualified Domain Name
5.8	(m) Attribut:	UUID/GUID-based Client Identifier
5.9	(m) Attribut:	Control DHCP for Address Resolution
6	(o) Attribut:	Device Initiative Option
6.1	(m) Attribut:	Device Initiative
7	(o) Attribut:	Manufacturer Specific Option
7.1	(m) Attribut:	List of Manufacturer Specific Suboptions
7.1.1	(m) Attribut:	Manufacturer Specific Suboption

7.1.2	(m) Attribut:	Manufacturer OUI
7.1.3	(m) Attribut:	Manufacturer Specific String
8	(m) Attribut:	Protocol machine Parameter
8.1	(m) Attribut:	Max Retry Limit
8.2	(m) Attribut:	UC Client Timeout
8.3	(m) Attribut:	MC Client Timeout
8.4	(m) Attribut:	TagControlInformation
8.4.1	(m) Attribut:	VLAN ID
8.4.2	(m) Attribut:	Priority
8.5	(m) Attribut:	Client Hold Time

SERVICES:

1	(m) OpsService:	Get
2	(m) OpsService:	Set
3	(m) OpsService:	Identify
4	(o) OpsService:	Hello

6.3.1.2.3 Attributs**Implicit**

L'attribut clé « Implicit » indique que l'objet DCP est adressé de façon implicite par le service.

List of Device Options

Cet attribut liste doit contenir toutes les options et sous-options qui sont prises en charge par l'appareil. Il agit comme un inventaire sur les options et sous-options de DCP prises en charge.

Device Option

Cet attribut contient une option qui est prise en charge par l'appareil.

Type d'attribut: Unsigned8

Valeurs permises: IP, DEVICE_PROPERTIES, DHCP, CONTROL, DEVICE_INITIATIVE, MANUFACTURER_SPECIFIC_128, ... MANUFACTURER_SPECIFIC_254

Device Suboption

Cet attribut contient une sous-option relative à une option spécifique qui est prise en charge par l'appareil.

Type d'attribut: Unsigned8

Valeurs permises:

Si Device Option = IP:

MAC_ADDRESS, IP_PARAMETER, FULL_IP_SUITE

Si Device Option = DEVICE_PROPERTIES:

DEVICE_VENDOR, NAME_OF_STATION, DEVICE_ID, DEVICE_ROLE,
DEVICE_OPTIONS, ALIAS_NAME, DEVICE_INSTANCE

Si Device Option = DHCP:

DHCP_PARAMETER

Si Device Option = CONTROL:

START, STOP, SIGNAL, FACTORY_RESET, RESET_TO_FACTORY

Si Device Option = DEVICE_INITIATIVE:

DEVICE_INITIATIVE_STATE

Si Device Option = MANUFACTURER_SPECIFIC:

MANUFACTURER_SPECIFIC_0, ... MANUFACTURER_SPECIFIC_255

IP Options

Cet attribut contient les attributs de sous-option suivants:

Adresse MAC

Cet attribut contient l'adresse physique de l'interface Ethernet conformément à l'adresse MAC de l'IEEE 802.3.

L'adresse MAC (MAC Address) doit être non volatile. Elle peut être prête au moyen du service Get ou elle peut être utilisée au sein du service Identify comme filtre. Elle ne doit pas être utilisée comme une sous-option au sein du service Set.

Type d'attribut: OctetString[6]

IP Parameter

L'attribut IP Parameter doit avoir la même signification que les attributs correspondants au sein de l'ASE "IP suite" (c'est-à-dire: suite IP) tel que défini en 6.3.13. Le paramètre de service Data Qualifier doit être utilisé pour adresser des valeurs d'attributs, utilisées actuellement ou stockées en permanence, pour les services DCP. Cette liste d'attributs contient les attributs suivants:

IP Address

Cet attribut contient l'adresse IP non volatile de l'interface Ethernet selon l'IETF RFC 791 et l'IETF RFC 3330

Type d'attribut: Unsigned32

Valeur par défaut: 0.0.0.0

Subnet Mask

Cet attribut contient le masque de sous-réseau non volatil de l'interface Ethernet selon l'IETF RFC 791 et l'IETF RFC 3330

Type d'attribut: Unsigned32

Valeur par défaut: 0.0.0.0

Standard Gateway

Cet attribut contient l'adresse IP non volatile de la passerelle normalisée selon l'IETF RFC 791 et l'IETF RFC 3330

Type d'attribut: Unsigned32

Valeur par défaut: 0.0.0.0

DNS Server Addresses

Cet attribut contient jusqu'à quatre adresses IP de serveur DNS. La valeur de "0.0.0.0" indique l'absence d'adresse.

Type d'attribut: Unsigned32[4]

Valeur par défaut: 0.0.0.0

Device Properties Option

Cet attribut contient les attributs de sous-options suivants:

Name Of Station

Cet attribut contient le nom de l'interface fourni par l'ingénierie. La valeur de cet attribut doit être la même que celle de l'attribut Chassis ID de l'ASE IEEE 802.1AB. Les valeurs "port-xyz" ou "port-xyz-rstuv", où x, y, z se situent dans la plage "0"-"9" de 001 jusqu'à 255 maximum et r, s, t, u, v se situent dans la plage "0"-"9" de 00000 jusqu'à 65535 maximum, ne doivent pas être utilisées.

NOTE 1 Bien garder à l'esprit que l'attribut Name Of Station est, contrairement à son nom trompeur (nom de station), un Name Of Interface (nom d'interface), car, comme énoncé ci-dessus, l'ASE DCP se focalise sur une interface Ethernet particulière.

Type d'attribut: VisibleString[240]

List of Ports

Cet attribut contient pour chacun des ports d'interface les attributs suivants.

Alias Name Of Station

Cet attribut contient les pseudonymes de l'interface dérivée des options de Link Layer Discovery Protocol (protocole de découverte de couche de liaison (LLDP)). Un pseudonyme particulier doit concaténer le the Peer Port ID, le "." et le Peer Chassis

ID. Il existe au maximum un seul pseudonyme pour chacun des ports. Une chaîne NULL (par exemple "") indique qu'aucun pseudonyme n'existe sur ce port.

L'attribut Alias Name Of Station peut seulement être utilisé comme filtre dans le service Identify. Les services Get, Set et Hello n'offrent pas d'accès à cet attribut.

NOTE 2 Exemple de pseudonyme Name Of Station = "port-001.mill-1.factory3.org".

Type d'attribut: VisibleString[255]

Device Role

Cet attribut contient une carte de bits pour indiquer les rôles de l'appareil sur cette interface Ethernet.

Les rôles de l'appareil indiquent les rôles possibles de l'appareil dans le contexte de PROFINET IO. Le rôle IO_DEVICE indique que l'interface Ethernet peut fournir jusqu'à une instance d'appareil E/S, le rôle IO_MULTIDEVICE indique qu'elle peut fournir plus d'une instance d'appareil E/S, le rôle IO_CONTROLLER indique qu'elle peut jouer le rôle d'appareil de commande E/S tandis que le rôle IO_SUPERVISOR indique qu'elle peut jouer le rôle de superviseur E/S.

Type d'attribut: Unsigned8

Valeurs permises: IO_DEVICE, IO_CONTROLLER, IO_MULTIDEVICE, IO_SUPERVISOR

Device Vendor

Cet attribut contient une chaîne spécifique à un vendeur. Il peut s'agir du type de l'appareil ou d'un numéro d'ordre.

Type d'attribut: VisibleString

Device ID; Identificateur d'appareil

Cet attribut contient le Device Ident Number (numéro d'identification d'appareil) tel que décrit en 7.6.1.

Type d'attribut: Unsigned32

Device Instance

Cet attribut contient le numéro d'instance ou de nœud de l'UUID d'objet de l'appareil d'E/S

Type d'attribut: Unsigned16

DHCP Option

Cet attribut facultatif contient les attributs de sous-options suivants:

Nom de l'hôte

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

Vendor Specific Information

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

Server Identifier

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

Parameter Request List

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

Class Identifier

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

DHCP Client Identifier

Cet attribut contient l'identificateur du client pour l'usage du protocole de configuration de serveur dynamique (DHCP)

Type d'attribut: voir IETF RFC 2132

Fully Qualified Domain Name

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

UUID/GUID-based Client Identifier

Cet attribut contient la valeur selon l'IETF RFC 2132.

Type d'attribut: voir IETF RFC 2132

Control DHCP for Address Resolution

Cet attribut contient des informations de commande pour l'usage du protocole de configuration de serveur dynamique (DHCP)

Type d'attribut: Unsigned8

Valeurs permises: DONT_USE_DHCP, DONT_USE_AND_RESET_DHCP_OPTIONS, USE_DHCP

Device Initiative Option

Cet attribut facultatif contient les attributs de sous-options suivants:

Device Initiative

Cet attribut contient la valeur pour activer ou désactiver l'émission du service Hello.

Type d'attribut: Unsigned16

Valeurs permises: ON, OFF

Manufacturer Specific Option

Cet attribut facultatif contient les attributs de sous-options suivants:

List of Manufacturer Specific Suboptions

Cette liste d'attributs contient les attributs suivants:

Manufacturer Specific Suboption

Cet attribut contient la valeur spécifique à un fabricant.

Type d'attribut: Unsigned8

Manufacturer OUI

Cet attribut contient un identificateur propre à une organisation pour identifier le vendeur.

Type d'attribut: OctetString[3]

Manufacturer Specific String

Cet attribut contient les informations individuelles.

Type d'attribut: OctetString

Protocol machine Parameter

Cet attribut contient les attributs suivants pour décrire le comportement des diagrammes de protocole:

Max Retry Limit

Cet attribut contient la limite maximale pour répéter un service client au cas où aucune réponse n'aurait été reçue par le client.

Type d'attribut: Unsigned8

Valeur par défaut: 4

Valeurs permises: 0 à 15

UC Client Timeout

Cet attribut contient la valeur maximale, en secondes, de la durée pendant laquelle un client attend une réponse en monodiffusion.

Type d'attribut: Unsigned16

Valeur par défaut: 1

Valeurs permises: 1 à 30

MC Client Timeout

Cet attribut contient la valeur maximale, en millisecondes, de la durée pendant laquelle un client attend toutes les réponses d'identification possibles. La valeur doit être supérieure à la durée Response Delay Time («délai de réponse») calculée à l'aide du Response Delay Factor («facteur de délai de réponse») issu de la demande. Elle doit être de 400 ms si le Response Delay Factor est égal à un et, autrement, arrondie par excès à la seconde entière plus 1 s.

Type d'attribut: Unsigned16

Valeurs permises: 400 à 65 000

Tag Control Information

Cet attribut contient les attributs suivants:

VLAN ID

Cet attribut contient le VLAN ID selon l'IEEE 802.1Q.

Type d'attribut: Unsigned16

Valeur permise: 0

Priority

Cet attribut contient l'étiquette de priorité de la trame selon l'IEEE 802.1Q.

Type d'attribut: Unsigned8

Valeur permise: 0

Client Hold Time

Cet attribut contient la valeur maximale, en secondes, qu'un serveur doit seulement accorder aux services Get ou Set du dernier client actif. Les demandes de services issues d'un autre serveur doivent être ignorées avant.

Type d'attribut: Unsigned16

Valeur par défaut: 3

6.3.1.3 Spécification de services de DCP**6.3.1.3.1 Get**

Le service Get est utilisé pour lire un ou plusieurs attributs d'ASE au moyen du modèle de communications client/serveur. Le groupe d'attributs doit être adressé au moyen d'une option. Le sous-groupe doit être adressé au moyen d'une sous-option.

Le serveur doit répondre avec le statut individuel "not supported" (c'est-à-dire: non pris en charge) si des options ou sous-options facultatives ne sont pas disponibles. D'une manière générale, les données demandées doivent recevoir une réponse tant qu'elles s'inscrivent dans la DCP-Get-ResPDU. Autrement, le statut "resource error" (c'est-à-dire: erreur de ressource) pour la première sous-option inadaptée doit être la réponse. Si l'option demandée avec la sous-option contient une liste contenant plus d'un élément, tous les éléments disponibles de la liste doivent recevoir réponse tant qu'ils s'inscrivent dans l'unité de données de protocole (PDU). L'ordre des éléments est arbitraire. Le Tableau 37 montre les paramètres du service.

Tableau 37 – Get

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
DA	M	M(=)		
List Of Options	M	M(=)		
Option	M	M(=)		
Suboption	M	M(=)		

Nom de paramètre	req	ind	rsp	cnf
Result(+)			S	S(=)
List Of Data			M	M(=)
Option			M	M(=)
Suboption			M	M(=)
Longueur			M	M(=)
Status			M	M(=)
IP Info			C	C(=)
Data (Données)			C	C(=)
AddUserData			C	C(=)
Result (-)			S	S(=)
ERRCLS			M	M(=)
ERRCODE			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

DA

Ce paramètre doit contenir l'adresse MAC IEEE 802.3 individuelle du serveur.

List Of Options

Ce paramètre contient la liste demandée avec des options accompagnées de leurs sous-options connexes. Ces paramètres doivent correspondre aux attributs de l'ASE.

Option

Ce paramètre contient l'option qui doit être lue dans l'appareil. Il peut contenir les attributs d'ASE suivants: IP, Device Properties, DHCP, et Manufacturer Specific. L'option Control ne doit pas être utilisée au sein de ce service. En outre, l'option All Selector peut être demandée comme option proxy pour récupérer toutes les informations disponibles auprès du serveur. Dans ce cas, la réponse doit au moins contenir l'option IP avec la sous-option IP Parameter et l'option Device Parameter avec les sous-options Device ID, Device Role, Device Options, et Name Of Station. La réponse peut contenir des options/sous-options complémentaires.

Suboption

Ce paramètre contient la sous-option correspondant à l'option qui doit être lue dans l'appareil. Selon l'option utilisée, seules doivent être utilisées les sous-options désignées qui sont relatives aux attributs d'ASE groupés.

Result (+)

Ce paramètre indique que la demande de service a réussi Il doit contenir les valeurs demandées ou un code d'erreur si une option/sous-option n'est pas disponible.

List Of Data

Ce paramètre doit contenir toutes les valeurs des options demandées avec les sous-options.

Option

Ce paramètre contient l'option demandée au à l'appareil.

Suboption

Ce paramètre contient la sous-option demandée à l'appareil.

Length

Ce paramètre contient le nombre d'octets du paramètre Data et du paramètre facultatif AddUserData selon le paramètre Status de la réponse de service.

Status

Ce paramètre contient le statut du paramètre Data.

Valeurs permises: NO_ERROR– données demandées livrées en totalité,
 OPTION_NOT_SUPPORTED– l'Option facultative demandée n'est pas prise en charge,
 SUBOPTION_NOT_SUPPORTED – la sous-option facultative demandée n'est pas prise en charge,
 RESSOURCE_ERROR - les données disponibles demandées ne s'inscrivent pas complètement dans la DCP-Get-ResPDU

Le statut RESSOURCE_ERROR indique que la valeur d'une option n'a pas pu être livrée en totalité si elle a dépassé la taille disponible de la PDU. Cependant, il convient que les données soient présentes tant qu'elles ont un sens du point de vue de l'application. Par exemple, si les éléments du paramètre List of Alias Names dans son ensemble dépassent toutefois la taille de la PDU, il convient que les éléments qui s'inscrivent déjà soient présents pour livrer au moins partiellement les informations demandées.

IP Info

Ce paramètre facultatif contient des informations complémentaires dans le cas de la sous-option IP Parameter et le paramètre Status contient la valeur NO_ERROR.

Valeurs permises: IP_VIA_SET – IP Parameter via service Set reçu,
 IP_VIA_DHCP – IP Parameter via service DHCP reçu,
 IP_NOT_ACTIVE – IP Parameter non actif, conflit d'adresse détecté

Data

Ce paramètre contient la valeur de l'attribut d'ASE demandé tant qu'il s'inscrit complètement dans la DCP-Get-ResPDU. Ce paramètre doit seulement être présent si le Status contient la valeur NO_ERROR.

AddUserData

Ce paramètre facultatif peut contenir des données spécifiques à un vendeur dans un format spécifique à un vendeur. Ce paramètre de réponse facultatif peut seulement être présent si le Status contient une valeur autre que NO_ERROR.

Result(-)

Ce paramètre indique que la demande de service a échoué en général pour des raisons locales.

ERRCLS

Le paramètre ERRCLS contient la classe d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: CTXT – contexte local non valide, PROTOCOL – erreur d'émission

ERRCODE

Le paramètre ERRCODE contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: INVALID_STATE si ERRCLS=CTX, LMPM si ERRCLS=PROTOCOL (erreur locale au cours de l'émission), TIMEOUT si ERRCLS=PROTOCOL (la station distante ne répondait pas)

6.3.1.3.2 Set

Le service Set est utilisé pour écrire un ou plusieurs attributs d'ASE au moyen du modèle de communications client/serveur. Le groupe d'attributs doit être adressé au moyen d'une option. Le sous-groupe doit être adressé au moyen d'une sous-option.

En outre, des commandes de contrôle spéciales commandent le comportement du serveur. Les commandes Start Transaction et Stop Transaction doivent être utilisées pour regrouper une séquence de service Set et le serveur doit être verrouillé pour un client dans ce cas. La commande Signal permet à sa sous-option FlashOnce d'identifier visuellement l'adresse matérielle. La commande Reset To Factory Settings (Réinitialiser aux réglages d'usine) permet avec ses sous-options de réinitialiser en particulier les données d'application, les

paramètres de communication, les paramètres d'ingénierie, tous les paramètres, l'appareil entier ou le micrologiciel ("firmware").

Le serveur doit répondre avec le statut individuel "not supported" (c'est-à-dire: non pris en charge) si des options ou sous-options facultatives ne sont pas disponibles. L'ordre des éléments est arbitraire.

Le Tableau 38 montre les paramètres du service.

Tableau 38 – Set

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
DA	M	M(=)		
List Of Data	U	U(=)		
Option	M	M(=)		
Suboption	M	M(=)		
Longueur	M	M(=)		
Data Qualifier	M	M(=)		
Manufacturer OUI	U	U(=)		
Data (Données)	M	M(=)		
List Of Control Commands	U	U(=)		
Control	M	M(=)		
Start Transaction	S	S(=)		
Stop Transaction	S	S(=)		
Factory Reset Settings	U	U(=)		
Reset To Factory Settings	U	U(=)		
Signal	U	U(=)		
FlashOnce	M	M(=)		
Result(+)			S	S(=)
List Of Response			M	M(=)
Option			M	M(=)
Suboption			M	M(=)
Length			M	M(=)
Status			M	M(=)
Add User Data			C	C(=)
Result (-)			S	S
ERRCLS			M	M
ERRCODE			M	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

DA

Ce paramètre doit contenir l'adresse MAC IEEE 802.3 individuelle du serveur.

List Of Data

Ce paramètre contient les données devant être écrites avec des options accompagnées de leurs sous-options connexes. Ces paramètres doivent correspondre aux attributs de l'ASE.

Option

Ce paramètre contient l'option qui doit être écrite dans le serveur. Il peut contenir les attributs d'ASE suivants: IP, Device Properties, DHCP, et Manufacturer Specific. L'option IP avec la sous-option MAC Address ne doit pas être utilisée.

Suboption

Ce paramètre contient la sous-option correspondant à l'option qui doit être écrite dans le serveur. Selon l'option utilisée, seules doivent être utilisées les sous-options désignées qui sont relatives aux attributs d'ASE groupés.

Length

Ce paramètre contient le nombre d'octets du paramètre.

Data Qualifier

Ce paramètre contient une information complémentaire des données qui doivent être écrites dans le serveur. En fonction de l'option/sous-option demandée, le statut a différentes significations.

Valeurs permises pour Option IP:
USE_TEMPORARY_AND_CLEAR_STORED_ADDRESS, SAVE_PERMANENT

Valeurs permises pour autre Option: USE_TEMPORARY, SAVE_PERMANENT

Manufacturer OUI

Ce paramètre facultatif contient l'identificateur OUI pour l'option spécifique à un fabricant.

Data

Ce paramètre contient les données qui doivent être écrites dans le serveur.

List Of Control Commands

Ce paramètre contient le paramètre suivant:

Control

Ce paramètre contient le paramètre sous-option suivant:

Start Transaction

Ce paramètre doit marquer le début d'une séquence de différents services Set. Le serveur doit verrouiller les attributs d'ASE pour le client demandeur jusqu'à ce que la commande Stop Transaction soit indiquée ou qu'une certaine temporisation se produise.

Type d'attribut: Unsigned8

Valeur permise: START

Stop Transaction

Ce paramètre doit marquer la fin d'une séquence de différents services Set. Le serveur doit déverrouiller les attributs d'ASE.

Type d'attribut: Unsigned8

Valeur permise: STOP

Factory Reset Settings

Ce paramètre doit être utilisé pour réinitialiser tous les attributs de DCP au moyen des services Set. Le comportement sur l'application de l'appareil n'est pas spécifié. Pour ces raisons, ce paramètre est obsolète et remplacé par le Reset To Factory Settings en dessous.

Type d'attribut: Unsigned8

Valeur permise: FACTORY_RESET

Reset To Factory Settings

Ce paramètre doit être utilisé pour réinitialiser l'appareil selon le qualificatif spécifié. Noter que les questions juridiques et les questions spécifiques à une application sont définies par les profils et par l'application locale.

Type d'attribut: Unsigned8

Valeur permise: RESET_FACTORY_SETTINGS

Reset application data

Réinitialiser à des valeurs d'usine les données qui ont été stockées en permanence dans des sous-modules et modules

- Les Record Data (données d'enregistrement) spécifiques à un fabricant doivent être mises aux valeurs d'usine
- Toutes les valeurs I&M1 à I&M15 doivent être mises aux valeurs d'usine.

La prise en charge de cette sous-option est recommandée.

Type d'attribut: Unsigned16

Valeur permise: RESET_APPLICATION_DATA

Reset communication parameters

Tous les paramètres actifs pour l'interface ou les ports et les AR doivent être mis aux valeurs par défaut et s'ils sont stockés de manière permanente, réinitialisés.

Ce mode est destiné à mettre l'interface de communication adressée d'un appareil dans un état presque similaire à l'état «out of the box» (en sortie de boîtier).

- NameOfStation doit être mis à "".
- Le paramètre IP suite doit être mis à 0.0.0.0
- Les paramètres DHCP (si disponibles) doivent être réinitialisés.
- Tous les paramètres PDev (PD IR Data, PD Port Data Adjust, PD Interface MRP Data Adjust...) doivent être mis aux valeurs d'usine.

La prise en charge de cette sous-option est obligatoire.

Type d'attribut: Unsigned16

Valeur permise: RESET_COMMUNICATION_PARAMETER

Reset engineering parameters

Réinitialisent aux valeurs d'usine les paramètres d'ingénierie qui avaient été stockés de façon permanente dans l'IOC ou dans l'IOD.

- Si un nœud peut faire passer sa fonctionnalité de IOC à IOD et vice-versa par l'intermédiaire du système d'ingénierie, la fonctionnalité d'usine doit être activée.
- Un programme d'application chargé par un système d'ingénierie doit être réinitialisé (ou supprimé).
- Une configuration chargée par un système d'ingénierie doit être réinitialisée (ou supprimée).

La prise en charge de cette sous-option est recommandée.

Type d'attribut: Unsigned16

Valeur permise: RESET_ENGINEERING_PARAMETER

Si un IOC ou IOD ne possède pas ce genre de paramètres, il doit répondre par un success.

Reset all parameters

Réinitialise à leurs valeurs d'usine dans l'IOD ou dans l'IOC toutes les données stockées.

- Si un nœud peut faire passer sa fonctionnalité de IOC à IOD et vice-versa par l'intermédiaire du système d'ingénierie, la fonctionnalité d'usine doit être activée.
- Un programme d'application chargé par un système d'ingénierie doit être réinitialisé (ou supprimé).
- Une configuration chargée par un système d'ingénierie doit être réinitialisée (ou supprimée).

La prise en charge de cette sous-option est recommandée.

Type d'attribut: Unsigned16

Valeur permise: RESET_ALL_PARAMETER

NOTE 1 Cela couvre toutes les données relatives à RESET_APPLICATION_DATA, RESET_COMMUNICATION_PARAMETER and RESET_ENGINEERING_PARAMETER.

Reset device

Réinitialise à leurs valeurs d'usine dans l'IOD ou dans l'IOC toutes les données stockées.

Ce service doit réinitialiser les paramètres de communication de toutes les interfaces de l'appareil et il convient qu'il réinitialise tous les paramètres de l'appareil.

Ce mode est destiné à mettre l'appareil dans un état similaire à l'état « out of the box » (en sortie de boîtier).

La prise en charge de cette sous-option est obligatoire.

Type d'attribut: Unsigned16

Valeur permise: RESET_DEVICE

NOTE 2 Même comportement que RESET_COMMUNICATION_PARAMETER, mais couvre aussi d'autres interfaces, et en plus (si pris en charge) même comportement que RESET_ALL_PARAMETERS.

Reset firmware

Réinitialiser les révisions logicielles installées aux images d'usine.

La prise en charge de cette sous-option est facultative.

Type d'attribut: Unsigned16

Valeur permise: RESET_FIRMWARE

Signal

Ce paramètre contient le paramètre suivant:

Flash Once

Ce paramètre doit contenir la valeur du signal. "Flash Once" (c'est-à-dire: clignoter une fois) signifie faire clignoter la DEL Ethernet LINK (ou signalisation de remplacement) avec une durée de 3 s à une fréquence de 1 Hz (500 ms allumée, 500 ms éteinte).

Type d'attribut: Unsigned16

Valeur permise: FLASH_ONCE

Result (+)

Ce paramètre indique que la demande de service a réussi Il doit contenir les valeurs demandées.

List Of Response

Ce paramètre contient les valeurs de retour de l'opération d'écriture ("write") de l'option/sous-option demandée.

Option

Ce paramètre contient l'option demandée à l'appareil.

Suboption

Ce paramètre contient la sous-option demandée à l'appareil.

Length

Ce paramètre contient le nombre d'octets du paramètre Data et du paramètre facultatif AddUserData selon le paramètre Status de la réponse de service.

Status

Ce paramètre contient le statut de transport du paramètre Data. La valeur NO_ERROR signale la livraison et, si demandé, le stockage permanent des données. Il peut s'écouler un certain temps avant que le paramètre ne fonctionne.

Valeurs permises: NO_ERROR– données de demande livrées en totalité,
OPTION_NOT_SUPPORTED– l'Option facultative demandée n'est pas prise en charge,
SUBOPTION_NOT_SUPPORTED–la sous-option facultative demandée n'est pas prise en charge,
SET_NOT_POSSIBLE – établissement ("set") pas possible pour des raisons locales,
IN_OPERATION – établissement ("set") pas possible à cause de l'application en fonctionnement

AddUserData

Ce paramètre facultatif peut contenir des données spécifiques à un vendeur dans un format spécifique à un vendeur. Ce paramètre de réponse facultatif peut seulement être présent si le Status contient une valeur autre que NO_ERROR.

Result(-)

Ce paramètre indique que la demande de service a échoué en général pour des raisons locales.

ERRCLS

Le paramètre ERRCLS contient la classe d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: CTXT – contexte local non valide, PROTOCOL – erreur d'émission

ERRCODE

Le paramètre ERRCODE contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: INVALID_STATE si ERRCLS=CTX, LMPM si ERRCLS=PROTOCOL (erreur locale au cours de l'émission), TIMEOUT si ERRCLS=PROTOCOL (la station distante ne répondait pas)

6.3.1.3.3 Local Set Command

Le service local Set Command est utilisé pour indiquer un service Set à la FAL, si l'attribut de l'élément est mis à **QUERY** (voir 6.3.1.2.1).

Le service transmet les arguments du service Set; aussi, pour une description plus détaillée sur les arguments, voir 6.3.1.3.2.

Le Tableau 39 montre les paramètres du service.

Tableau 39 – Local Set Command

Nom de paramètre	ind	rsp
Argument	M	
DA	M	
List Of Data	U	
Option	M	
Suboption	M	
Longueur	M	
Data Qualifier	M	

Nom de paramètre	ind	rsp
Manufacturer OUI	U	
Data	M	
List Of Control Commands	U	
Commande	M	
Start Transaction	S	
Stop Transaction	S	
Factory Reset Settings	U	
Reset To Factory Settings	U	
Signal	U	
FlashOnce	M	
Result(+)		S
List Of Response		M
Option		M
Suboption		M
Longueur		M
Status		M
Add User Data		C
Result (-)		S
ERRCLS		M
ERRCODE		M

6.3.1.3.4 Identify

Le service Identify peut être utilisé pour découvrir des appareils sur le réseau. Le service doit être envoyé vers une adresse MAC IEEE 802.3 en multidiffusion bien connue qui est définie par le protocole DCP. Seuls les appareils qui satisfont au filtre demandé doivent répondre à cette demande. La confirmation de Identify doit contenir toutes les réponses reçues dans les limites d'un certain intervalle de temps. Les appareils répondants sont identifiés par l'adresse MAC source IEEE 802.3. Autrement, la confirmation doit contenir une liste vide.

Si les critères de filtrage sont remplis, un serveur doit répondre avec les données pour les options et sous-options qui sont parties intégrantes du filtre. En outre, les données d'option

- IP avec la sous-option IP Parameter et ses éléments IP Address, Subnet Mask, et Standard Gateway,
- Device Properties avec ses sous-options List of Station Names, Device ID, Device Role, et Device Option

doivent toujours être incluses dans la réponse. La séquence de blocs de données peut être arbitraire. Le Tableau 40 montre les paramètres du service.

Tableau 40 – Identify

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
List Of Filter	S	S(=)		
Option	M	M(=)		
Suboption	M	M(=)		

Nom de paramètre	req	ind	rsp	cnf
Length	M	M(=)		
Data	M	M(=)		
All Selector	S	S(=)		
Response Delay Factor	M	M(=)		
Result(+)			S	S(=)
List of Devices			C	C(=)
SA				M
List of Data			M	M(=)
Option			M	M(=)
Suboption			M	M(=)
Length			M	M(=)
Data			C	C(=)
Result (-)			S	S(=)
ERRCLS			M	M(=)
ERRCODE			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

List Of Filter

Ce paramètre conditionnel contient les options/sous-options qui doivent être utilisées pour filtrer la demande côté serveur. Plus d'un filtre doit être traité avec une opération AND (c'est-à-dire: ET logique) du côté serveur.

NOTE L'opération AND signifie qu'un serveur répond seulement à une demande de service Identify dans les limites du délai de réponse si tous les critères de filtrage concordent avec les valeurs d'attributs actuellement utilisées.

Option

Ce paramètre contient la valeur de l'option qui doit être utilisée pour filtrer la demande côté serveur. Il peut contenir les attributs d'ASE suivants: IP, Device Properties, DHCP, et Manufacturer Specific.

Suboption

Ce paramètre contient la valeur de la sous-option qui doit être utilisée pour filtrer la demande côté serveur. Selon l'option utilisée, seules doivent être utilisées les sous-options désignées qui sont relatives aux attributs d'ASE groupés.

Length

Ce paramètre contient la longueur des données de l'option et de la sous-option qui doivent être utilisées pour filtrer la demande côté serveur.

Data

Ce paramètre contient les données de l'option et de la sous-option qui doivent être utilisées pour filtrer la demande côté serveur.

All Selector

Ce paramètre conditionnel indique que tous les appareils doivent au moins répondre avec leurs options obligatoires.

Response Delay Factor

Ce paramètre contient le facteur pour que le serveur calcule la valeur dispersée pour retarder la réponse si les critères de List Of Filter sont remplis. La base de temps est de 10 ms. Le retard calculé du serveur doit être compris entre 10 ms et 64 s.

Type de paramètre: Unsigned16

Valeurs permises: 1 à 6 400

Valeurs par défaut: 1 (Appareil de commande E/S)

Result (+)

Ce paramètre indique que la demande de service a réussi. Il doit contenir les valeurs demandées.

List of Devices

Ce paramètre contient la liste des appareils retournés à partir de l'identification.

SA

Ce paramètre contient l'adresse MAC IEEE 802.3 individuelle de l'appareil.

List of Data

Ce paramètre contient les données de l'appareil spécifié par SA. Il est constitué des éléments suivants:

Option

Ce paramètre contient l'option pour identifier les données retournées.

Suboption

Ce paramètre contient la sous-option pour identifier les données retournées.

Length

Ce paramètre contient la longueur des données retournées.

Data

Ce paramètre contient les données appartenant à l'option et à la sous-option spécifiées.

Result(-)

Ce paramètre indique que la demande de service a échoué en général pour des raisons locales.

ERRCLS

Le paramètre ERRCLS contient la classe d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: CTXT – contexte local non valide, PROTOCOL – erreur d'émission

ERRCODE

Le paramètre ERRCODE contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: INVALID_STATE si ERRCLS=CTX, LMPM si ERRCLS=PROTOCOL (erreur locale au cours de l'émission), TIMEOUT si ERRCLS=PROTOCOL (la station distante ne répondait pas)

6.3.1.3.5 Hello

Le service Hello facultatif peut être utilisé pour annoncer la présence d'un appareil dans un réseau. Le service doit être envoyé vers une adresse MAC IEEE 802.3 en multidiffusion bien connue qui est définie par le protocole DCP.

Le service doit émettre le nombre demandé de DCP-Hello-ReqPDU dans les limites d'une certaine durée restreinte. L'intervalle Hello demandé doit être utilisé pour attendre avant chaque réémission.

Le service Hello est destiné à être utilisé pour accélérer la phase de démarrage; par conséquent, il convient que l'émission ne dure pas plus d'une centaine de millisecondes.

Le Tableau 41 montre les paramètres du service.

Tableau 41 – Hello

Nom de paramètre	req	ind	cnf
Argument	M	M(=)	
List of Data	M	M(=)	
Name of Station	M	M(=)	
IP Parameter	M	M(=)	
IP Address	M	M(=)	
Subnet Mask	M	M(=)	
Standard Gateway	M	M(=)	
Device ID	M	M(=)	
List of Device Options	M	M(=)	
Device Option	M	M(=)	
Device Suboption	M	M(=)	
Longueur	M	M(=)	
Data (Données)	M	M(=)	
Device Role	M	M(=)	
Number of DCP-Hello-ReqPDUs	M		
Initial Delay	M		
Hello Interval	M		
Result (+)			S(=)
Result (-)			S(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

List Of Data

Cette liste de paramètres contient les paramètres suivants:

Name Of Station

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

IP Parameter

Ce paramètre est constitué des paramètres suivants:

IP Address

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

Subnet Mask

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

Standard Gateway

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

Device ID; Identificateur d'appareil

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

List of Device Options

Cette liste de paramètres est constituée des paramètres suivants:

Device Option

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

Device Suboption

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

Length

Ce paramètre contient la longueur des données.

Data

Ce paramètre contient les données de l'option et de la sous-option.

Device Role

Ce paramètre contient la valeur de l'attribut d'ASE correspondant.

Number of DCP-Hello-ReqPDUs

Ce paramètre contient le nombre des DCP-Hello-ReqPDU qui doivent être émises avec la demande de service

Type de paramètre: Unsigned32

Valeurs permises: 1 à 15

Valeurs par défaut: 3

Initial Delay

Ce paramètre contient le temps de retard pour la première DCP-Hello-ReqPDU.

Base de temps: 1 ms

Type de paramètre: Unsigned32

Valeurs permises: 0, 50, 100, 500, 1000

Hello Interval

Ce paramètre contient l'intervalle de temps entre l'émission de la première DCP-Hello-ReqPDU et celle de chaque DCP-Hello-ReqPDU suivante.

Type de paramètre: Unsigned16

Valeurs permises: 30_MILLISECONDS, 50_MILLISECONDS, 100_MILLISECONDS, 300_MILLISECONDS, 500_MILLISECONDS, 1_SECONDS

Valeurs par défaut: 30_MILLISECONDS

Result (+)

Ce paramètre indique que la demande de service a réussi. Il doit contenir les valeurs demandées.

Result(-)

Ce paramètre indique que la demande de service a échoué en général pour des raisons locales.

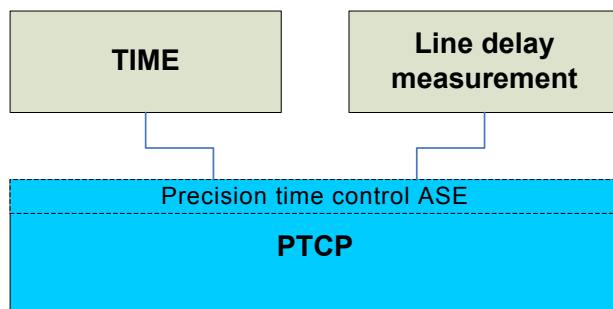
6.3.2 ASE "Precision time control"**6.3.2.1 Concepts**

Le protocole « Precision Transparent Clock Protocol » (PTCP) représente une méthode uniforme de distribution d'heure. Cela signifie en particulier qu'un maître (une mère) de PTCP synchronise un certain nombre d'esclaves de PTCP dans la gamme en dessous de la microseconde. Il prend également en charge le modèle de meilleure mère de PTCP et jusqu'à 32 domaines PTCP différents. Le protocole de transport PTCP est basé sur la couche liaison de données.

Un domaine PTCP doit être utilisé pour CLOCK, qui est une heure locale pour synchroniser l'accès réseau (en particulier la communication cyclique) dans tous les appareils qui participent au domaine PTCP particulier.

Aucun autre domaine PTCP n'est pris en charge à ce jour.

En outre, PTCP prend en charge la mesure du délai de ligne entre deux ports correspondants. La Figure 4 montre le contexte de l'ASE « Precision time control ».



Légende

Anglais	Français
Time	Temps
Line delay measurement.	Mesure de retard de ligne...
Precision time control ASE	ASE «commande de temps de précision»

Figure 4 – Applications de PTCP

6.3.2.2 Spécification de la classe PTCP

6.3.2.2.1 Modèle

ASE: ASE PTCP

CLASS: PTCP

CLASS ID: pas utilisé

PARENT CLASS: TOP

ATTRIBUTS:

- 1 (m) Attribut-clé: Sync ID
- 2 (m) Attribut: Domain
- 2.1 (m) Attribut: Domain UUID
- 2.2 (m) Attribut: Sequence ID
- 2.3 (m) Attribut: Sync Send Factor
- 2.4 (m) Attribut: PTCP Takeover Timeout
- 2.5 (m) Attribut: PTCP Timeout
- 2.6 (m) Attribut: Role (MASTER, SLAVE)
- 2.7 (c) Contrainte: Role = MASTER
- 2.7.1 (m) Attribut: Priority
- 2.7.2 (m) Attribut: Accuracy
- 2.7.3 (m) Attribut: Variance
- 2.7.4 (m) Attribut: PTCP Master Startup Time
- 2.8 (c) Contrainte: Sync ID = TIME
- 2.8.1 (m) Attribut: PTCP Time
- 2.8.1.1 (m) Attribut: Epoch Number
- 2.8.1.2 (m) Attribut: Seconds
- 2.8.1.3 (m) Attribut: Nano Seconds
- 2.8.1.4 (m) Attribut: Current UTC Offset
- 2.9 (m) Attribut: PLL Window
- 2.10 (m) Attribut: Sync State Info
- 2.10.1 (m) Attribut: Jitter out of Boundary
- 2.10.2 (m) Attribut: No Sync Message Received
- 2.10.3 (m) Attribut: Wrong PTCP Domain UUID
- 2.11 (m) Attribut: Rate Compensation Factor
- 2.12 (m) Attribut: List of Ports
- 2.12.1 (m) Attribut: PortID

2.12.2	(m) Attribut:	Borderline
2.12.2.1	(m) Attribut:	Ingress
2.12.2.2	(m) Attribut:	Egress
2.12.3	(m) Attribut:	Line Delay
2.12.4	(m) Attribut:	Rate Compensation Factor Peer
2.13	(o) Attribut:	List of OUIs
2.13.1	(m) Attribut:	OUI
2.13.2	(s) Attribut:	Subtype
2.13.3	(s) Attribut:	Data Block

SERVICES:

1	(m) OpsService:	Start Bridge
2	(m) OpsService:	Start Slave
3	(o) OpsService:	Start Master
4	(m) OpsService:	Stop Bridge
5	(m) OpsService:	Stop Slave
6	(o) OpsService:	Stop Master
7	(m) OpsService:	Sync State Change

6.3.2.2.2 Attributs**Sync ID**

Cet attribut-clé identifie l'instance du diagramme de protocole auquel les données appartiennent. La valeur CLOCK doit être utilisée pour la synchronisation de cycle.

Type d'attribut: Unsigned8

Valeurs permises: CLOCK

Domaine

Cet attribut est constitué des éléments suivants:

Domain UUID

Cet attribut contient l'UUID (identificateur unique universel) du domaine pour la synchronisation qui est fourni par le système d'ingénierie et émis par une mère. L'UUID est utilisé pour un regroupement logique d'horloges PTCP qui se synchronisent les unes aux autres à l'aide du protocole PTCP. Les horloges dans d'autres domaines PTCP ne doivent pas être synchronisées.

Type d'attribut: UUID

Sequence ID

Cet attribut contient le numéro de séquence du message de synchronisation (sync message). Il doit être incrémenté avec chaque nouveau message de synchronisation. C'est au moyen du Sequence ID que le destinataire détecte les doublons.

Type d'attribut: Unsigned16

Sync Send Factor

Cet attribut contient l'intervalle d'envoi pour les messages de synchronisation.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 64 000

PTCP Takeover Timeout

Cet attribut contient une valeur de temporisation pour détecter la perte de messages de synchronisation issus du maître de synchronisation actuel. L'esclave de synchronisation doit tenter de trouver un nouveau maître de synchronisation. La valeur de "PTCP takeover timeout" (temporisation de prise de contrôle PTCP) doit se situer dans la plage de 32 ms à 16 352 ms.

Type d'attribut: Unsigned32

Valeur par défaut: 96 ms

PTCP Timeout

Cet attribut contient une valeur de temporisation pour détecter la perte de tout message de synchronisation. La valeur de "PTCP timeout" (temporisation PTCP) doit se situer dans la plage de 32 ms à 16 352 ms.

Type d'attribut: Unsigned32

Valeur par défaut: 192 ms

Rôle

Cet attribut contient le rôle (rôle) dans la synchronisation du temps.

Type d'attribut: Unsigned16

Valeurs permises: MASTER, SLAVE

Priority

Cet attribut doit être la priorité du maître émettant les messages de synchronisation.

Type d'attribut: Unsigned8

Valeurs permises: PRIMARY, SECONDARY

Accuracy

Cet attribut doit être l'exactitude du temps de maître en comparaison à la précision de l'horloge locale.

Type d'attribut: Unsigned8

Valeurs permises: 25_ns, 100_ns, 250_ns, 1_μs, 2.5_μs, 10_μs, 25_μs, 100_μs, 250_μs, 1_ms, UNKNOWN

Valeur par défaut: 100_ns

Variance

Cet attribut doit être la valeur pour la qualité de l'horloge. Chaque horloge-mère doit maintenir une estimation de sa précision héritée. La variance de PTCP est basée sur la théorie de l'écart d'Allan. L'écart d'Allan est $\sigma_y(\tau)$ qui est estimé comme suit:

$$\sigma_y(\tau) = \left[\frac{1}{2(N-2)\tau^2} \sum_{k=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \right]^{\frac{1}{2}} \quad (15)$$

où

$\sigma_y(\tau)$ est l'écart d'Allan.

x_k , x_{k+1} et x_{k+2} sont des mesures résiduelles de temps, effectuées aux temps t_k , $t_k + \tau$ et $t_k + 2\tau$, entre le temps donné par l'horloge mesurée et une horloge de référence locale.

N est le nombre d'échantillons de données.

La variance PTCP est définie par:

$$\sigma_{PTCP}^2 = \tau^2 \times \frac{1}{3} \sigma_y^2 \quad (16)$$

où

σ_{PTCP}^2 est la variance de PTCP.

σ_y est l'écart d'Allan.

τ est la grandeur

Une estimation sans biais de la variance de PTCP doit être calculée comme suit:

$$\sigma_{PTCP}^2 = \frac{1}{3} \left[\frac{1}{2(N-2)} \times \sum_{k=1}^{N-2} (x_{k+2} - 2x_{k+1} + x_k)^2 \right] \quad (17)$$

où

σ_{PTCP}^2

est l'estimation sans biais de la variance de PTCP

x_k, x_{k+1} et x_{k+2}

sont des mesures résiduelles de temps, effectuées aux temps $t_k, t_k + \tau$ et $t_k + 2\tau$, entre le temps donné par l'horloge mesurée et une horloge de référence locale.

N

est le nombre d'échantillons de données.

Pour la variance de PTCP, la grandeur τ , la période d'échantillon, doit être la valeur définie comme intervalle de synchronisation.

Les variances de PTCP doivent être représentées comme suit:

Une estimation de la variance, σ_{PTCP}^2 , est calculée en unités de secondes carrées.

Le logarithme base 2 de cette estimation est calculée. Le calcul du logarithme peut ne pas être plus précis que la précision de l'estimation de la variance.

Le logarithme est multiplié par 28 pour produire une valeur normalisée.

La valeur est représentée comme un Integer16 en complément à deux. La valeur 0x8000 est ajoutée à la valeur rapportée représentée sous cette forme et tout dépassement de capacité est ignoré. Le résultat, à savoir la valeur rapportée normalisée de décalage, est transtypé ("cast") en Unsigned16.

Cette valeur normalisée décalée, représentée comme étant Unsigned16, doit être la valeur des log-variances.

Le nombre positif le plus grand possible, à savoir 0xFFFF, pour l'attribut log variance normalisée de décalage doit indiquer que la variance est soit trop grande pour être représentée, soit n'a pas été calculée. Les valeurs de variance sont utilisées dans la sélection de la meilleure horloge-mère.

Type d'attribut: Unsigned16

NOTE 1 Par exemple, supposons une valeur de la variance de PTCP de $1\,414 \times 2^{-73} = 1\,497 \times 10^{-22} \text{ s}^2$. Alors, $\log_2(1\,414 \times 2^{-73}) = -73 + 0,5 = -72,5$. Si cela était exprimé comme étant un Integer16, le résultat serait tronqué à -72. Pour conserver une certaine précision, la valeur est normalisée de 2^8 pour donner une valeur scaledLogVariance de -18 560, 0xB780, qui conserve une précision de 8 bits de plus. Il y est ajouté 0x8000 pour obtenir la valeur normalisée de décalage communiquée de 0x3780.

NOTE 2 La plus petite variance pouvant être représentée est 2^{-128} ou $\sim 3 \times 10^{-39} \text{ s}^2$, qui se traduit par une log variance normalisée de décalage de 0x0000. La variance maximale qui peut être représentée est $\sim 2^{+127,99609}$, qui se traduit par une log variance normalisée de décalage de 0xFFFF.

PTCP Master Startup Time

Cet attribut doit être la durée de la phase de démarrage du maître de synchronisation. Au cours de cette durée, le maître de synchronisation doit vérifier s'il y a un maître de synchronisation avec une meilleure qualité d'horloge dans le domaine de synchronisation.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 300 s

Valeur par défaut: 10 S

Time

Cet attribut est constitué des éléments suivants pour la représentation du temps:

PTCP Time

Cet attribut est constitué des éléments suivants pour la représentation du temps PTCP depuis le début de l'époque PTCP 1970-01-01T00:00:00Z.

Epoch Number

Cet attribut doit être le nombre courant de fois que l'horloge Seconds 32 bits est repassée par zéro depuis le début de l'époque PTCP.

Type d'attribut: Unsigned16

Seconds

Cet attribut contient le nombre courant de secondes depuis 1970-01-01T00:00:00Z.

Type d'attribut: Unsigned32

Nano Seconds

Cet attribut contient le nombre courant de nanosecondes jusqu'à la prochaine seconde.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 999 999 999

Current UTC Offset

Cet attribut contient le nombre courant de secondes intercalaires comme décalage de PTCP Time au Temps universel coordonnée (TUC).

Type d'attribut: Interger16

Clock

Cet attribut est constitué des éléments suivants pour la représentation d'horloge:

Local Time

Cet attribut consiste en une valeur d'heure locale en nanosecondes. Cette valeur doit être utilisée pour la synchronisation d'horloge PTCP et doit être la base pour le compteur de cycles.

Type d'attribut: Unsigned64

PLL Window

Cet attribut doit caractériser l'exactitude de synchronisation. L'écart entre l'heure locale et l'heure de référence émise par le maître ne doit pas dépasser la valeur de PLL Window.

Type d'attribut: Unsigned32

Sync State Info

Cet élément est constitué des attributs d'erreur suivants. La synchronisation a réalisé l'état synchrone s'il n'y a d'erreur.

Jitter out of Boundary

Cet attribut identifie si, oui ou non, l'exactitude de synchronisation nécessaire (à l'intérieur ou à l'extérieur de PLL Window) est obtenue.

Type d'attribut: Boolean

No Sync Message Received

Cet attribut identifie si, oui ou non, les messages de synchronisation ont été reçus.

Type d'attribut: Boolean

Wrong PTCP Domain UUID

Si un message de synchronisation avec le mauvais Domain UUID a été reçu, cet attribut sera mis.

Type d'attribut: Boolean

Rate Compensation Factor

Cet attribut doit être utilisé pour réajuster le temps de chevauchement à partir des messages de synchronisation et de retard.

Type d'attribut: Unsigned32

List of Ports

Cet attribut est constitué des éléments suivants:

PortID

Cet attribut identifie un Port d'un pont.

Type d'attribut: Unsigned8

Borderline

Les deux attributs qui suivent doivent contenir la valeur ON pour les ports d'extrémité d'un PTCP Domain. La valeur de l'attribut Ingress doit être ON et la valeur pour l'attribut Egress doit être OFF pour un port couplant deux segments PTCP, pour le segment supérieur hiérarchique. La valeur de l'attribut Ingress doit être OFF et la valeur pour l'attribut Egress doit être ON pour un port couplant deux segments PTCP, pour le segment inférieur hiérarchique.

NOTE 3 Il vise à éviter l'émission de messages de synchronisation PTCP et d'annonce PTCP depuis un segment inférieur hiérarchique vers un segment supérieur au sien d'un domaine PTCP.

Cet attribut est constitué des éléments suivants:

Ingress

Cet attribut identifie les ports d'extrémité d'un PTCP Domain.

Type d'attribut: Unsigned8

Valeurs permises: ON, OFF

Egress

Cet attribut identifie les ports d'extrémité d'un PTCP Domain ou une ligne frontière niveau/segment.

Type d'attribut: Unsigned8

Valeurs permises: ON, OFF

Line Delay

Cet attribut contient la latence d'émission de ce port vers le port connecté.

Type d'attribut: Unsigned32

Rate Compensation Factor Peer

Cet attribut contient le rapport entre les fréquences de quartz locale et distante de l'appareil au port connecté.

Type d'attribut: Float64

List of OUIs

Cet attribut est constitué des éléments suivants:

OUI

Cet attribut est un Organizationally Unique Identifier (c'est-à-dire: identificateur propre à une organisation) tel que défini par l'IANA.

Type d'attribut: Octet[3]

Subtype

Cet attribut est un paramètre spécifique à une organisation qui est valide dans le contexte de l'attribut OUI.

Type d'attribut: Unsigned8

Data Block

Cet attribut est un paramètre spécifique à une organisation qui est valide dans le contexte des attributs OUI et Subtype.

Type d'attribut: UUID

6.3.2.3 Spécification de service de PTCP**6.3.2.3.1 Start bridge**

Le service "Start bridge" crée une instance du diagramme du protocole PTCP. Le Tableau 42 montre les paramètres du service.

Tableau 42 – Start bridge

Nom de paramètre	req	cnf
Argument	M	
Sync ID	M	
List of Port Parameter	U	
Port ID	M	
Borderline	M	
Result(+)		S
Sync ID		M
Result (-)		S
Sync ID		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

List of Port Parameter

Cette liste facultative de paramètres est constituée des paramètres suivants:

Port ID

Ce paramètre contient l'ID du port qui émet le message de demande de retard.

Borderline

Ce paramètre contient les ports d'extrémité d'un domaine d'horloges.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Error code

Le paramètre Error code contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: ENTRY_NOT_POSSIBLE, SYNC_ID_EXISTS

6.3.2.3.2 Start slave

Le service "Start slave" ajoute des propriétés de synchronisation au diagramme du protocole PTCP with. Le Tableau 43 montre les paramètres du service.

Tableau 43 – Start slave

Nom de paramètre	req	cnf
Argument	M	
SyncID	M	
Subdomain UUID	M	
Sync Send Factor	M	
PTCP Timeout Factor	M	
PLL Window	M	
Result(+)		S
Sync ID		M
Result (-)		S
Sync ID		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Subdomain UUID

Ce paramètre contient l'UUID du sous-domaine de synchronisation. L'UUID est utilisé pour un regroupement logique d'horloges PTCP qui se synchronisent les unes aux autres à l'aide du protocole PTCP.

Sync Send Factor

Ce paramètre contient l'intervalle de réception des messages de synchronisation.

PTCP Timeout Factor

Ce paramètre contient une valeur de temporisation pour détecter la perte de messages de synchronisation.

PLL Window

Ce paramètre contient l'exactitude de synchronisation requise. L'écart entre l'heure locale et l'heure de référence émise par une grand-mère ne doit pas dépasser la valeur de PLL Window.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Error code

Le paramètre Error code contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: SLAVE_NOT_POSSIBLE, SLAVE_EXISTS

6.3.2.3.3 Start master

Le service "Start master" ajoute des propriétés de synchronisation avec la qualification pour devenir maître de synchronisation au diagramme du protocole PTCP. Le Tableau 44 montre les paramètres du service.

Tableau 44 – Start master

Nom de paramètre	req	cnf
Argument	M	
Sync ID	M	
Subdomain UUID	M	
Sync Send Factor	M	
Sync Class	M	
Clock Stratum	M	
Current UTC Offset	M	
Epoch Number	M	
Result(+)		S
Sync ID		M
Result (-)		S
Sync ID		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Subdomain UUID

Ce paramètre contient l'UUID du sous-domaine de synchronisation. L'UUID est utilisé pour un regroupement logique d'horloges PTCP qui se synchronisent les unes aux autres à l'aide du protocole PTCP.

Sync Send Factor

Ce paramètre contient l'intervalle d'envoi des messages de synchronisation.

Sync Class

Ce paramètre contient le rôle recommandé (primaire, secondaire) comme maître.

Clock Stratum

Ce paramètre contient la qualité du temps propre.

Current UTC Offset

Ce paramètre contient le décalage courant par rapport au TUC.

Epoch Number

Ce paramètre contient le nombre courant de fois que l'horloge Seconds 32 bits est repassée par zéro depuis le début de l'époque PTCP. L'époque PTCP comment à 0 heures le 1 janvier 1970.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Error code

Le paramètre Error code contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: MASTER_NOT_POSSIBLE, MASTER_EXISTS, MASTER_NOT_EXISTS

6.3.2.3.4 Stop bridge

Ce service doit être utilisé pour arrêter le diagramme du protocole PTCP. Le Tableau 45 montre le paramètre du service.

Tableau 45 – Stop bridge

Nom de paramètre	req	chf
Argument	M	
Sync ID	M	
Result(+)		S
Sync ID		M
Result (-)		S
Sync ID		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Error code

Le paramètre ErrCode contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: SYNC_ID_NOT_EXISTS

6.3.2.3.5 Stop slave

Ce service doit être utilisé pour arrêter le diagramme du protocole PTCP. La synchronisation sera arrêtée et la fonctionnalité de pontage demeure. Le Tableau 46 montre le paramètre du service.

Tableau 46 – Stop slave

Nom de paramètre	req	cnf
Argument	M	
Sync ID	M	
Result(+)		S
Sync ID		M
Result (-)		S
Sync ID		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result(-)

Ce paramètre indique que la demande de service a échoué.

SyncID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Error code

Le paramètre Error code contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: SLAVE_NOT_EXISTS

6.3.2.3.6 Stop master

Ce service doit être utilisé pour réduire les propriétés du diagramme du protocole PTCP. La fonctionnalité de maître sera retirée et le diagramme de protocole est forcé de prendre le rôle d'esclave. Le Tableau 47 montre le paramètre du service.

Tableau 47 – Stop master

Nom de paramètre	req	cnf
Argument	M	
Sync ID	M	
Result(+)		S
Sync ID		M
Result (-)		S
Sync ID		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Error code

Le paramètre Error code contient le code d'erreur de l'erreur spécifique.

Type: Unsigned16

Valeurs permises: MASTER_NOT_EXISTS

6.3.2.3.7 Sync state change

Ce service doit être utilisé pour indiquer un changement de l'état de synchronisation (à savoir "synchronized" (synchronisé) ou "out of synchronization" (hors synchronisation) et/ou un changement de la qualité de synchronisation (à savoir "Jitter out of Boundary" (gigue hors limite), "No Sync Message Received" (aucun message de synchronisation reçu)). Un changement peut seulement apparaître pour un esclave de synchronisation. Le Tableau 48 montre les paramètres du service.

Tableau 48 – Sync state change

Nom de paramètre	ind
Argument	M
Sync ID	M
Sync State	M
Reason List	C

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Sync ID

Il est l'attribut-clé pour identifier l'instance du diagramme de protocole.

Sync State

Cet attribut doit contenir la valeur SYNCHRONIZED ou NOT_SYNCHRONIZED.

Reason List

Cet attribut conditionnel est constitué des éléments suivants:

ErrorType

Cet attribut identifie une erreur de synchronisation.

Type d'attribut: Unsigned16

Valeurs permises: JITTER_OUT_OF_BOUNDARY, NO_SYNC_MESSAGE_RECEIVED, WRONG_PTCP_DOMAIN

Appear

Cet attribut identifie si l'erreur apparaît ou disparaît.

Type d'attribut: Boolean

Valeurs permises: TRUE, FALSE

6.3.2.3.8 Line Delay change

Ce service local doit être utilisé pour indiquer un changement de certaines Local Systems Data (données de systèmes locaux). Le Tableau 49 montre les paramètres du service.

Tableau 49 – Line Delay change

Nom de paramètre	ind
Argument	M
Local Port ID	M
Local Line Delay	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

Local Port ID

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Line Delay

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

6.3.2.3.9 Comportement de PTCP

6.3.2.3.9.1 Synchronization

La synchronisation est faite avec une trame de synchronisation et une trame de suivi facultative. La trame de suivi est nécessaire si le matériel ne peut pas ajouter des retards de pont et de ligne à la volée.

L'horloge émettant les trames de synchronisation dans un sous-domaine de PTCP est appelée maître/mère de PTCP.

Les horloges qui reçoivent des trames de synchronisation et réajuste leur heure par rapport à la mère de PTCP sont appelées "esclaves de PTCP". Leur exactitude de synchronisation dépend de l'exactitude des unités d'horodatage issues de chaque pont qui mesure les retards de ligne et de pont.

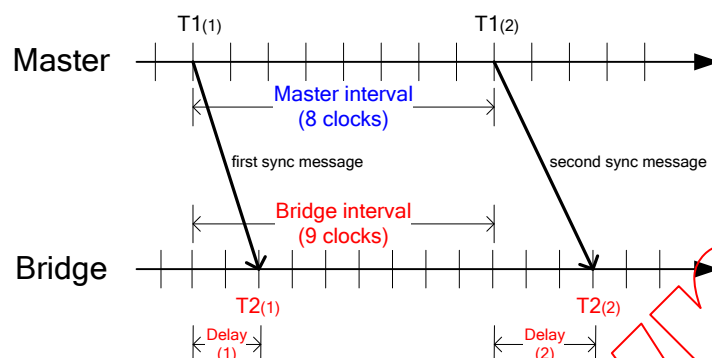
Un pont avec les améliorations suivantes est appelé "horloge transparente de PTCP" (PTCP transparent clock):

- Mesure de retards de ligne
- Mesure de retard de pont
- Compensation de fréquence pour les retards de ligne et de pont
- Mécanisme de prise en charge de la redondance des supports pour PTCP

6.3.2.3.9.2 Compensation de commande de fréquence

La compensation de commande de fréquence égalise la fréquence d'horloge d'une horloge transparente PTCP à celle du maître de PTCP. C'est la raison pour laquelle les retards de pont et de ligne cumulés, reçus par l'esclave de PTSP pour assurer une exactitude de synchronisation plus élevée, sont mesurés avec la fréquence d'horloge du maître de PTCP.

Une horloge transparente PTCP utilise des trames de synchronisation pour mesurer la fréquence d'horloge relative au maître de PTCP. La Figure 5 montre la mesure de dérive d'horloge.



Légende

Anglais	Français
Master	Maître (mère)
Master interval (8 clocks)	Intervalle de maître (8 horloges)
Bridge	Pont
Bridge interval (9 clocks)	Intervalle de pont (9 horloges)
First sync message	Premier message de synchronisation
Second sync message	Second message de synchronisation
Delay	Retard

Figure 5 – Mesure de dérive d'horloge

$$\text{RateCompFactor} = [T1_{(2)} - T1_{(1)}] / [(T2_{(2)} - \text{Delay}(2)) - (T2_{(1)} - \text{Delay}(1))] \quad (18)$$

où

<i>RateCompFactor</i>	est le facteur de compensation de fréquence
$T1_{(1)}$	est l'instant de début du premier message de synchronisation issu du Master (maître/mère)
$T1_{(2)}$	est l'instant de début du deuxième message de synchronisation issu du Master (maître/mère)
$T2_{(1)}$	est l'instant d'arrivée du premier message de synchronisation sur le Bridge (pont)
$T2_{(2)}$	est l'instant d'arrivée du deuxième message de synchronisation sur le Bridge (pont)
$\text{Delay}(1)$	est la différence temporelle du premier message de synchronisation entre l'instant de début sur le Master et l'instant d'arrivée sur le Bridge
$\text{Delay}(2)$	est la différence temporelle du deuxième message de synchronisation entre l'instant de début sur le Master et l'instant d'arrivée sur le Bridge

L'horloge transparente PTCP utilise le facteur de compensation de fréquence pour corriger les valeurs mesurées des temps résidentiels.

$$\text{CompBridgeDelay} = \text{RateCompFactor} * \text{BridgeDelay} \quad (19)$$

où

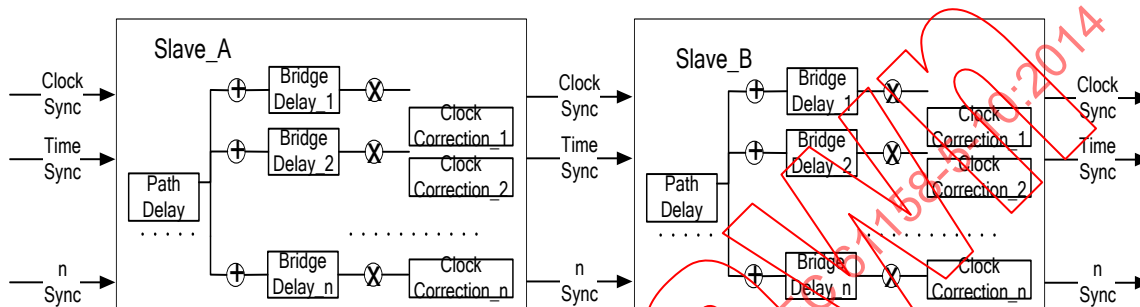
<i>CompBridgeDelay</i>	est le retard de pont compensé
<i>RateCompFactor</i>	est le facteur de compensation de fréquence

BridgeDelay

est le retard du pont

6.3.2.3.9.3 Synchronisation plurielle

Le protocole PTCP doit être utilisé pour la synchronisation d'horloge, la synchronisation de temps ainsi que d'autres sortes de synchronisation et il prend en charge jusqu'à 32 domaines de synchronisation. Il peut y avoir diverses sources auxquelles toutes les autres horloges se synchronisent. Les sous-domaines ne sont pas synchronisés au niveau de PTCP. Chaque esclave de PTCP doit se synchroniser lui-même. La mesure de retard doit être effectuée pour chaque voie de communication. La prise en charge de la synchronisation d'horloge et de la synchronisation de temps est obligatoire. La Figure 6 montre une synchronisation plurielle



Légende

Anglais	Français
Slave A	Esclave A
Slave B	Esclave B
Clock Sync	Synchronisation d'horloge
Time Sync	Synchronisation de temps
m Sync	Synchronisation m
Path Delay	Retard de voie
Bridge Delay	Retard de pont
Clock connection	Connexion d'horloge

Figure 6 – Synchronisation plurielle

La correction d'horloge (Clock Correction) est constituée du décalage de fréquence et de l'erreur de dérive des oscillateurs. Chaque horloge transparente (Transparent Clock) doit corriger le temps de résidence des messages de synchronisation et de retard avec la valeur correspondante de Clock Correction.

6.3.2.3.9.4 Best Master Algorithm (Algorithme de meilleure mère)

Le Best Master Algorithm (BMA) spécifie un algorithme indiquant comment une horloge détermine l'horloge préférentielle émettant des messages de synchronisation. L'algorithme tourne de façon indépendante sur chaque horloge (mère/esclave) dans un système PTCP. Chaque horloge prend sa propre décision pour la meilleure mère.

Pour calculer la meilleure mère, une horloge doit:

- considérer la qualité des plus récents messages de synchronisation reçus;
- inclure les résultats obtenus dans le précédent calcul de meilleure mère. Cependant, s'il y a des messages de synchronisation qualifiés plus récents, les valeurs issues de ces messages doivent être prises en considération.

Pour analyser la qualité d'un message de synchronisation, une horloge doit prendre en considération les éléments suivants:

- la mère émettant le message de synchronisation appartient à son propre sous-domaine,
- la réception périodique des messages de synchronisation,
- la strate de l'horloge-mère,
- la variance qui décrit la caractéristique de l'horloge-mère.

Si la strate et la variance de mères différentes sont égales, le BMA utilise l'adresse MAC de la source pour décider de l'horloge préférentielle (la plus petite gagne).

6.3.2.4 Prise en charge de la redondance de supports pour la synchronisation avec PTCP

Pour améliorer la fiabilité d'un réseau homologue à homologue ("peer-to-peer"), des voies de communication redondantes sont nécessaires. Des voies de communication redondantes peuvent conduire à des trames en circulation. Afin d'éviter les trames en circulation, les voies de communication redondantes ne sont pas utilisées pour des messages de données.

Les messages de synchronisation PTCP utilisent les voies de communication redondantes pour être indépendants vis-à-vis de la reconfiguration de réseaux redondants et pour de courtes voies de communication.

L'algorithme de transmission suivant doit être utilisé pour éviter les messages de synchronisation en circulation dans un réseau avec des voies de communication redondantes. L'algorithme de transmission pour PTCP veille à ce qu'un seul message de synchronisation soit émis sur chaque voie de communication.

Le premier message de synchronisation reçu est appelé "message de synchronisation primaire" (primary Sync message). Le message de synchronisation primaire doit être utilisé pour la synchronisation.

6.3.2.5 Invocation de l'objet PTCP

Pour l'invocation de l'objet PTCP, les règles suivantes s'appliquent:

- Des objets PTCP doivent exister pour chaque Sync ID.

6.3.3 ASE Media redundancy

6.3.3.1 Vue d'ensemble

Pour la redondance des supports, la CEI 62439-2 s'applique. La prise en charge de la base d'information de gestion (MIB) MRP est facultative.

L'ASE Media redundancy (redondance de supports) spécifie une méthode uniforme d'obtention d'une redondance de supports avec temps de rétablissement déterministe dans le cas d'une seule défaillance de réseau (défaillance de liaison, défaillance de nœud de commutateur provoquant des défaillances de liaison).

NOTE 1 La redondance de supports fonctionne de manière transparente pour l'application. Aucun changement de l'application n'est exigé.

Le Media Redundancy Protocol (MRP «Protocole de redondance de supports) est conçu pour réagir de façon déterministe sur une seule défaillance d'une liaison de jonction ou d'un nœud de jonction du réseau.

Un réseau conforme doit avoir une topologie en anneau avec plusieurs nœuds.

L'un des nœuds a le rôle d'un gestionnaire de redondance de supports (media redundancy manager (MRM)). La fonction du MRM est d'observer et commander la topologie en anneau afin de réagir à des pannes du réseau. Le MRM le fait en envoyant des trames sur un des ports d'anneau sur l'anneau et les reçoit de l'anneau sur son autre port d'anneau, et vice versa dans l'autre sens.

Les autres nœuds dans l'anneau ont le rôle de clients de redondance de supports (media redundancy clients (MRC)). Un MRC réagit sur des trames de reconfiguration issues du MRM et peut signaler des changements de liaison sur ses ports d'anneau.

Un nœud conforme doit avoir la capacité de devenir un MRM ou un MRC, ou les deux. Chaque nœud nécessite un commutateur intégré avec au moins deux ports (ports d'anneau) connectés à l'anneau.

Chaque nœud doit être capable de détecter la défaillance ou le rétablissement d'une liaison de jonction ou la défaillance ou le rétablissement d'un nœud voisin.

Le protocole MRP est constitué d'une entité service et d'une entité protocole. Elles sont décrites toutes les deux dans la CEI 62439-2.

6.3.3.2 Spécification de la classe Media redundancy

6.3.3.2.1 Généralités

L'ASE "Media redundancy" définit un type d'objet.

6.3.3.2.2 Modèle

Un objet Media redundancy est décrit par le modèle suivant:

ASE:	ASE Media redundancy
CLASS:	Media redundancy
CLASS ID:	pas utilisé
PARENT CLASS:	IEEE 802.3 IEEE 802.1D IEEE 802.1 AB
ATTRIBUTS:	
1	(m) Attribut-clé: MRP Domain UUID
2	(m) Attribut: MRP Domain Name
3	(m) Attribut: MRP Version
4	(m) Attribut: MRP Role (MANAGER, CLIENT)
5	(m) Attribut: MRP Params
5.1	(c) Contrainte: MRP Role = MANAGER
5.1.1	(m) Attribut: Manager Priority
5.1.2	(m) Attribut: Topology Change Interval
5.1.3	(m) Attribut: Topology Change Repeat Count
5.1.4	(m) Attribut: Short Test Interval
5.1.5	(m) Attribut: Default Test Interval
5.1.6	(m) Attribut: Test Monitoring Count
5.2	(c) Contrainte: MRP Role = CLIENT
5.2.1	(m) Attribut: Link Down Interval
5.2.2	(m) Attribut: Link Up Interval
5.2.3	(m) Attribut: Link Change Count
6	(m) Attribut: MRP Data Check
6.1	(m) Attribut: MRP Domain UUID
6.2	(m) Attribut: MRP Check
6.2.1	(m) Attribut: Multiple Manager Check
6.2.2	(m) Attribut: Domain UUID Check

7	(m) Attribut:	Ring State Data
8	(m) Attribut:	Ring Port 1 ID
9	(m) Attribut:	Ring Port 2 ID
10	(o) Attribut:	VLAN ID

SERVICES:

aucun

6.3.3.2.3 Attributs**MRP Domain UUID**

Cet attribut-clé définit le domaine de redondance représentant l'anneau auquel l'objet Media redundancy appartient. Il est mis à la valeur par défaut de l'ID de domaine ou fourni comme ID unique par ingénierie.

Type d'attribut: UUID

MRP Domain Name

Cet attribut définit le domaine de redondance représentant l'anneau auquel l'objet Media redundancy appartient. Il est mis à la valeur par défaut du nom de domaine ou fourni comme ID unique par ingénierie.

Type d'attribut: voir Chassis ID

MRP Version

Cet attribut spécifie la version de l'objet Media redundancy dans le domaine de redondance référencé par la valeur de l'attribut Domain ID.

Type d'attribut: Unsigned16

Valeurs permises: 1

MRP Role

Cet attribut spécifie le rôle de l'objet Media redundancy dans le domaine de redondance référencé par la valeur de l'attribut Domain ID.

Type d'attribut: Unsigned16

Valeurs permises: MANAGER, CLIENT

MRP Params

Cet attribut spécifie les paramètres de l'objet Media redundancy dans le domaine de redondance référencé par la valeur de l'attribut Domain ID.

Il est constitué des attributs suivants:

MRP Prio

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "MANAGER". Cet attribut contient la priorité du gestionnaire. Une valeur inférieure indique une plus haute priorité, 0x0000 (plus haute priorité) à 0xF000 (plus faible priorité) par incréments de 0x1000.

Type d'attribut: Unsigned16

MRP Topology Change Interval

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "MANAGER". Cet attribut contient l'intervalle pour envoyer des trames de changement de topologie.

Type d'attribut: Unsigned16

MRP Topology Change Repeat Count

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "MANAGER". Cet attribut contient le compte d'intervalles qui commande l'émission répétée des trames de changement de topologie.

Type d'attribut: Unsigned16

MRP Short Test Interval

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "MANAGER". Cet attribut contient l'intervalle court pour envoyer des trames d'essai sur des ports d'anneau après des changements de liaisons dans l'anneau.

Type d'attribut: Unsigned16

MRP Default Test Interval

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "MANAGER". Cet attribut contient l'intervalle par défaut pour envoyer des trames d'essai sur des ports d'anneau.

Type d'attribut: Unsigned16

MRP Test Monitoring Count

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "MANAGER". Cet attribut contient le compte d'intervalles pour surveiller la réception des trames d'essai.

Type d'attribut: Unsigned16

MRP Link Down Interval

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "CLIENT". Cet attribut contient l'intervalle pour envoyer des trames Link Change (changement de liaison) "link down" (liaison perdue) sur des ports d'anneau.

Type d'attribut: Unsigned16

MRP Link Up Interval

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "CLIENT". Cet attribut contient l'intervalle pour envoyer des trames Link Change (changement de liaison) "link up" (liaison établie) sur des ports d'anneau.

Type d'attribut: Unsigned16

MRP Link Change Count

Cet attribut est présent lorsque la valeur de l'attribut "MRP role" est "CLIENT". Cet attribut contient le compte de trames Link Change qui commande l'émission répétée des trames Link up (liaison établie) ou Link down (liaison perdue).

Type d'attribut: Unsigned16

MRP Data Check

Ce paramètre contient les règles de vérification. Il est constitué des attributs suivants:

MRP Domain UUID

Ce paramètre contient l'UUID de domaine MRP (MRP Domain UUID) par rapport auquel effectuer la vérification, si l'attribut Domain UUID Check est mis à TRUE.

Type d'attribut: UUID

MRP Check

Cet attribut est constitué des éléments suivants:

Multiple Manager Check

Cet attribut contient la valeur TRUE pour activer l'évaluation continue de plusieurs gestionnaires MRP avec la réception des MRP-PDU avec MRP_Type=MRP_Test. Autrement, il doit être mis à FALSE.

Type d'attribut: Boolean

Domain UUID Check

Cet attribut contient la valeur TRUE pour activer l'évaluation continue de MRP Domain UUID avec chaque indication Remote System Data Change telle que spécifiée dans le diagramme d'états Neighborhood Check. Autrement, il doit être mis à FALSE.

Type d'attribut: Boolean

Ring State Data

Cet attribut contient l'état d'anneau réel de l'objet Media redundancy dans le domaine de redondance.

Type d'attribut: Unsigned16

Valeurs permises:

- OPEN (L'anneau est ouvert en raison de la défaillance de liaison ou d'appareil dans l'anneau),
- CLOSED (L'anneau est fermé (fonctionnement normal, aucune erreur)),
- UNDEFINED (Doit être mis si l'attribut Real Role State contient la valeur CLIENT (à savoir l'objet Media redundancy était reconfiguré au rôle de client))

Ring Port 1 ID

Cet attribut spécifie un port d'un pont qui est affecté comme ring port 1 (port d'anneau 1) dans le domaine de redondance référencé par la valeur de l'attribut Domain ID. Un port doit seulement appartenir à un seul domaine de redondance.

Type d'attribut: Unsigned16

Ring Port 2 ID

Cet attribut spécifie un autre port d'un pont différent de Ring Port 1 ID qui est affecté comme port d'anneau 2 (ring port 2) dans le domaine de redondance référencé par la valeur de l'attribut Domain ID.

Type d'attribut: Unsigned16

VLAN ID

Cet attribut facultatif peut être utilisé par l'objet Media redundancy et spécifie son identificateur de VLAN dans le domaine de redondance.

Type d'attribut: Unsigned16

6.3.3.3 Spécification des services Media redundancy

La spécification de services de redondance de supports est définie dans la CEI 62439-2.

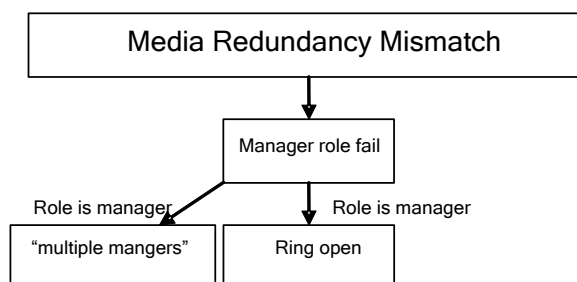
6.3.3.4 Comportement de la redondance de supports avec diagnostics et alarmes

6.3.3.4.1 Généralités

Les événements de redondance de supports doivent provoquer des événements de diagnostic et des notifications d'alarmes si une relation d'applications est établie et l'attribut Check Media Redundancy a la valeur TRUE.

6.3.3.4.2 Dépendances du diagnostic de redondance de supports

La Figure 7 montre les dépendances du diagnostic de redondance de supports:



Légende

Anglais	Français
Media redundancy mismatch	Discordance de redondance de supports

Anglais	Français
Manager role fails	Le rôle de Responsable/Gestionnaire a échoué
Role is manager	Le rôle est «Responsable/Gestionnaire»
multiple managers	Plusieurs Responsables/Gestionnaires
Ring open	Anneau ouvert

Figure 7 – Dépendances du diagnostic de redondance de supports

Si un nœud dans un domaine de redondance n'est pas dans un rôle de gestionnaire, un événement de diagnostic "Manager role fail" (c'est-à-dire: échec de rôle de gestionnaire) doit être signalé. Tous les autres événements de diagnostic de redondance de supports sont verrouillés.

Si un nœud dans un domaine de redondance est dans un rôle de gestionnaire et un autre gestionnaire est détecté, l'événement "multiple managers" (plusieurs gestionnaires) doit être signalé. Cet événement peut se produire parallèlement à l'événement d'état d'anneau « ring open » (anneau ouvert).

Si un nœud dans un domaine de redondance est dans un rôle de gestionnaire et un anneau ouvert est détecté, l'événement "ring open" doit être signalé.

Ces événements sont indiqués au moyen du service State Change.

De plus, la redondance de supports traite l'événement suivant pour les deux ports d'anneau d'un domaine de redondance au moyen du service Neighborhood Change:

Si l'attribut Check Ring Port Neighborhood est TRUE et l'attribut Ring Port Domain State est BAD ou UNDEFINED, l'événement "Peer Domain ID mismatch" (discordance d'ID de domaine d'homologues) doit être signalé.

6.3.4 ASE Real time cyclic

6.3.4.1 Vue d'ensemble

Le protocole temps réel cyclique (RTC) est un concept d'écriture et de lecture de données cycliques au moyen de services tamponnés ("bufférisés"). Il utilise les services de l'IEEE 802.3 ou du protocole datagramme d'utilisateur (UDP).

6.3.4.2 Spécification de la classe RTC

6.3.4.2.1 Généralités

L'ASE "RTC" définit pour chaque relation de communication (CR) particulière un type d'objet RTC.

6.3.4.2.2 Modèle

Un objet RTC est décrit par le modèle suivant:

ASE:	ASE RTC
CLASS:	RTC
CLASS ID:	pas utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Attribut-clé: CREP
2	(m) Attribut: Protocol machine Parameter
2.1	(m) Attribut: DA

2.2	(m) Attribut:	SA
2.3	(m) Attribut:	Frame ID
2.4	(m) Attribut:	Protocol Machine
2.5	(c) Contrainte:	Protocol Machine = PPM
2.5.1	(m) Attribut:	VLAN Prio
2.5.2	(m) Attribut:	VLAN ID
2.5.3	(m) Attribut:	Tx Option
2.5.3.1	(m) Attribut:	RT Class
2.5.3.2	(c) Contrainte:	RT Class = RT_CLASS_3
2.5.3.2.1	(m) Attribut:	Frame Send Offset
2.5.3.2.2	(m) Attribut:	Tx Port
2.5.3.2.3	(m) Attribut:	Startup Mode
2.5.3.2.4	(m) Attribut:	Media Redundancy
2.5.3.2.5	(c) Contrainte:	Media Redundancy = TRUE
2.5.3.2.5.1	(m) Attribut:	Redundant Frame ID
2.5.3.2.5.2	(m) Attribut:	Redundant Frame Send Offset
2.5.3.2.5.3	(m) Attribut:	Redundant Tx Port
2.5.3.2.6	(m) Attribut:	DFP
2.5.3.2.7	(c) Contrainte:	DFP = TRUE
2.5.3.2.7.3	(m) Attribut:	Sub Frame Position
2.5.4	(m) Attribut:	Reduction Ratio
2.5.5	(m) Attribut:	Phase
2.5.6	(m) Attribut:	Séquence
2.6	(c) Contrainte:	Protocol Machine = CPM
2.6.1	(m) Attribut:	Rx Option
2.6.1.1	(m) Attribut:	RT Class
2.6.1.2	(c) Contrainte:	RT Class = RT_CLASS_3
2.6.1.2.1	(m) Attribut:	Startup Mode
2.6.1.2.2	(m) Attribut:	Media Redundancy
2.6.1.2.3	(c) Contrainte:	Media Redundancy = TRUE
2.6.1.2.3.1	(m) Attribut:	Redundant Frame ID
2.6.1.2.3.2	(m) Attribut:	Redundancy Watchdog
2.6.1.2.4	(m) Attribut:	DFP
2.6.1.2.5	(c) Contrainte:	DFP = TRUE
2.6.1.2.5.1	(m) Attribut:	Sub Frame Position
2.6.2	(m) Attribut:	Data Hold Factor

SERVICES:

1	(m) OpsService:	PPM Set Prov Data
2	(m) OpsService:	PPM Set Prov Status
3	(m) OpsService:	PPM Activate
4	(m) OpsService:	PPM Close
5	(m) OpsService:	PPM Start
6	(m) OpsService:	PPM Error
7	(m) OpsService:	CPM Get Cons Data
8	(m) OpsService:	CPM Get Cons Status
9	(m) OpsService:	CPM Set RedRole
10	(m) OpsService:	CPM Activate
11	(m) OpsService:	CPM NoData
12	(m) OpsService:	CPM Stop

6.3.4.2.3 Attributs**CREP**

Cet attribut-clé contient l'identificateur de la CR dans la classe CRL

Type d'attribut: Unsigned32

Protocol machine Parameter

Cet attribut contient les attributs suivants pour décrire le comportement des diagrammes de protocole:

DA

Ce paramètre contient l'adresse de destination (c'est-à-dire l'adresse du consommateur) selon l'IEEE 802.3.

Type d'attribut: OctetString[6]

SA

Ce paramètre contient l'adresse de la source (c'est-à-dire l'adresse du fournisseur) selon l'IEEE 802.3.

Type d'attribut: OctetString[6]

Frame ID

Cet attribut contient l'identificateur de la trame.

Type d'attribut: Unsigned16

Valeurs permises: voir Tableau 276

Protocol Machine

Cet attribut contient le diagramme de protocole particulier sélectionné pour cette CR. Ce diagramme de protocole est mis à PPM (diagramme de protocole fournisseur cyclique), lorsque l'appareil est le fournisseur de la CR particulière. Autrement, il est mis à CPM (diagramme de protocole client cyclique), lorsque l'appareil est le client de la CR particulière.

Type d'attribut: Unsigned8

Valeur permise: PPM, CPM

VLAN Prio

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est "PPM". Il contient la priorité de la trame selon l'IEEE 802.1Q.

Type d'attribut: Unsigned8

VLAN ID

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est "PPM". Il contient le VLAN ID selon l'IEEE 802.1Q.

Type d'attribut: Unsigned16

Valeur permise: 0

Tx Option

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est "PPM". Il spécifie les options d'émission du fournisseur et il est constitué des attributs suivants:

RT Class

Cet attribut contient la classe de communication du fournisseur.

Type d'attribut: Unsigned32

Valeurs permises: RT_CLASS_1, RT_CLASS_2, RT_CLASS_3, RT_CLASS_UDP

NOTE 1 La valeur RT_CLASS_2 est utilisable dans le contexte de démarrage hérité seulement.

Frame Send Offset

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il contient le décalage d'envoi relatif par rapport au début du rapport de réduction et de la phase connexes. La base de temps est de 1 ns. La valeur doit être inférieure à la durée d'une phase.

Type d'attribut: Unsigned32

Valeurs permises: BEST_EFFORT – (obligatoire) dès que possible, 0-0x003D08FF pour la programmation optimisée

Tx Port

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique le port d'émission, où la trame doit être envoyée.

Type d'attribut: Unsigned8

Valeurs permises: 0x01 à 0xFF

Startup Mode

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique le mode de démarrage pour RT_CLASS_3.

Valeurs permises: Legacy, Advanced

Media Redundancy

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'utilisation de la MRPD (redondance de supports avec duplication planifiée des trames).

Valeurs permises: FALSE, TRUE

Redundant Frame ID

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient l'identificateur de la trame redondante.

Type d'attribut: Unsigned16

Valeurs permises: voir Tableau 276

Redundant Frame Send Offset

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient le décalage d'envoi relatif par rapport au début du rapport de réduction et de la phase connexes. La base de temps est de 1 ns. La valeur doit être inférieure à la durée d'une phase.

Type d'attribut: Unsigned32

Valeurs permises: BEST EFFORT – (obligatoire) dès que possible, 0 - 0x003D08FF pour la programmation optimisée

Redundant Tx Port

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il indique le port, où la trame redondante doit être envoyée.

Type d'attribut: Unsigned8

Valeurs permises: 0x01 à 0xFF

DFP

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'usage de DFP.

Valeurs permises: FALSE, TRUE

Sub Frame Position

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "DFP" est "TRUE". Il définit le numéro de position de la sous-trame. Les numéros de position doivent être dans l'ordre strictement croissant au sein de la liste de données de sous-trames (List of Sub frame Data).

Type d'attribut: Unsigned8

Valeurs permises: 0x01 à 0x3F pour emballer et déemballer les sous-trames pendant le transport et de bout en bout; 0x40 à 0x7F pour emballer et déemballer les sous-trames de bout en bout; 0x00 et 0x80 à 0xFF réservés

Reduction Ratio

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est "PPM". Il contient le rapport de réduction de la CR. Le cycle effectif d'envoi est déterminé par Reduction Ratio multiplié par l'horloge d'envoi.

Type d'attribut: Unsigned16

Valeurs permises: voir Tableau 277

Phase

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est "PPM". Il indique le cycle particulier d'envoi pendant lequel les données doivent être envoyées. Ce paramètre dépend du paramètre de rapport de réduction (Reduction Ratio).

Type d'attribut: Unsigned16

Valeurs permises: 1 à la valeur sélectionnée du Reduction Ratio

Séquence

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est "PPM". Il contient la position de la trame de données dans la file d'attente d'envoi. La valeur 0 doit être utilisée pour indiquer une séquence indéterminée.

Type d'attribut: Unsigned16

Rx Option

Cet attribut conditionnel est présent lorsque la valeur de l'attribut du diagramme de protocole est « CPM ». Il spécifie les options de réception du consommateur et est constitué des attributs suivants:

RT Class

Cet attribut contient la classe de communication du consommateur.

Type d'attribut: Unsigned32

Valeurs permises: RT_CLASS_1, RT_CLASS_2, RT_CLASS_3, RT_CLASS_UDP

NOTE 2 La valeur RT_CLASS_2 est utilisable dans le contexte de démarrage hérité seulement.

Startup Mode

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique le mode de démarrage de l'AR correspondante.

Valeurs permises: Legacy, Advanced

Media Redundancy

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'utilisation de MRPD.

Valeurs permises: FALSE, TRUE

Redundant Frame ID

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est "TRUE". Il contient l'identificateur de la trame redondante.

Type d'attribut: Unsigned16

Valeurs permises: voir Tableau 276

Redundancy Watchdog Timer

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient la valeur du temporisateur chien de garde de redondance (redundancy watchdog timer).

Type d'attribut: Unsigned16

Base de temps: 1 s

Valeurs permises: 0x0001 à 0x0100

Valeur par défaut: 0x0001

NOTE 3 Le «redundancy watchdog» (chien de garde de redondance) est censé détecter les problèmes d'anneau «statique» (par exemple: un câble est tiré). Ainsi, la résolution du «redundancy watchdog timer» est relâchée.

DFP

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'usage de DFP.

Valeurs permises: FALSE, TRUE

Sub Frame Position

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "DFP" est "TRUE". Il définit le numéro de position de la sous-trame. Les numéros de position doivent être dans l'ordre strictement croissant au sein de la liste de données de sous-frames (List of Sub frame Data).

Type d'attribut: Unsigned8

Valeurs permises: 0x01 à 0x3F pour empaqueter et décompacter les sous-frames pendant le transport et de bout en bout; 0x40 à 0x7F pour empaqueter et décompacter les sous-frames de bout en bout; 0x00 et 0x80 à 0xFF réservés

Data Hold Factor

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "protocol machine" est «CPM». Il contient un facteur qui est utilisé pour calculer le Data Hold Time (temps de conservation des données). Ce temps est l'intervalle de temps pour retenir les dernières données de sortie si aucune donnée E/S n'a été reçue.

L'intervalle pour le Data Hold Time est le produit des valeurs des attributs Data Hold Factor, Send Clock Factor, Reduction Ratio et de la base de temps de 31,25 µs relatifs au consommateur surveillé. Le Data Hold Time doit être inférieur ou égal à 1,92 s si la valeur de l'attribut "RT Class" de l'AR est 0x01, 0x02 ou 0x03 et il doit être inférieur ou égal à 49,152 s si la valeur de l'attribut "RT Class" de l'AR est 0x04.

Type d'attribut: Unsigned16

Valeurs permises: 0x0001 à 0x1E00 pour une valeur de "RT Class" de 0x01, 0x02 ou 0x03; 0x0001 à 0xF000 pour une valeur de "RT Class" de 0x04

6.3.4.3 Spécification du service RTC**6.3.4.3.1 PPM Set Prov Data**

Ce service local est utilisé pour mettre à jour le tampon d'émission du fournisseur avec de nouvelles données. Le Tableau 50 montre les paramètres du service.

Tableau 50 – PPM Set Prov Data

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Data	M	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Data

Ce paramètre contient les données pour mettre à jour la mémoire tampon d'émission.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.2 PPM Set Prov Status

Ce service local est utilisé pour mettre à jour le statut APDU du tampon d'émission du fournisseur. Le Tableau 51 montre les paramètres du service.

Tableau 51 – PPM Set Prov Status

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Status	M	
Result(+)		S
CREP		M

Nom de paramètre	req	cnf
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Status

Ce paramètre contient l'état APDU pour mettre à jour la mémoire tampon d'émission.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.3 PPM Activate

Ce service local est utilisé pour activer le diagramme de protocole du fournisseur (PPM). Le Tableau 52 montre les paramètres du service.

Tableau 52 – PPM Activate

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
VLAN Prio	M	
VLAN ID	M	
Tx Option	M	
RT Class	M	
Frame Send Offset	C	
Tx Port	C	
Startup Mode	C	
Media Redundancy	C	

Nom de paramètre	req	cnf
Redundant Frame ID	C	
Redundant Frame Offset	C	
Redundant Frame Tx Port	C	
DFP	C	
Sub Frame Position	C	
Reduction Ratio	M	
Phase	M	
Séquence	M	
Default Values	M	
Default Status	M	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

DA

Ce paramètre contient l'adresse MAC de destination devant être utilisée pour la trame RTC.

SA

Ce paramètre contient l'adresse MAC source devant être utilisée pour la trame RTC.

Frame ID

Ce paramètre contient le Frame ID devant être utilisé pour la trame RTC.

VLAN Prio

Ce paramètre contient la VLAN Prio devant être utilisée pour la trame RTC.

VLAN ID

Ce paramètre contient le VLAN ID devant être utilisé pour la trame RTC.

Tx Option

Cet attribut définit l'option d'émission. Il se compose des éléments suivants:

RT Class

Cet attribut contient la classe de communication du fournisseur.

Frame Send Offset

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il contient le décalage d'envoi relatif par rapport au début du rapport de réduction et de la phase connexes.

Tx Port

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique le port d'émission, où la trame doit être envoyée.

Startup Mode

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique le mode de démarrage de l'AR correspondante.

Media Redundancy

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'utilisation de MRPD.

Redundant Frame ID

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient l'identificateur de la trame redondante.

Redundant Frame Send Offset

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient le décalage d'envoi relatif par rapport au début du rapport de réduction et de la phase connexes.

Redundant Tx Port

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il indique le port, où la trame redondante doit être envoyée.

DFP

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'usage de DFP.

Sub Frame Position

Cet attribut conditionnel est présent lorsque la valeur de l'attribut DFP est "TRUE". Il définit le numéro de position de la sous-trame.

Reduction Ratio

Ce paramètre contient le Reduction Ratio à utiliser pour la trame RTC.

Phase

Ce paramètre contient la Phase devant être utilisée pour la trame RTC.

Séquence

Ce paramètre contient la Sequence devant être utilisée pour la trame RTC.

Default Value

Ce paramètre contient les données initiales pour tampon d'émission.

Default Status

Ce paramètre contient l'état initial de l'APDU pour le tampon d'émission.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTEXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.4 PPM Close

Ce service local est utilisé pour fermer le diagramme de protocole du fournisseur (PPM) et, donc, arrêter l'émission. Le Tableau 53 montre les paramètres du service.

Tableau 53 – PPM Close

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		M
CREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

6.3.4.3.5 PPM Start

Ce service local est utilisé pour lancer l'émission du diagramme de protocole du fournisseur (PPM). Le Tableau 54 montre les paramètres du service.

Tableau 54 – PPM Start

Nom de paramètre	ind
Argument	M
CREP	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

6.3.4.3.6 PPM Error

Ce service local est utilisé pour indiquer une erreur au cours de l'émission d'un diagramme de protocole de fournisseur (PPM). Le Tableau 55 montre les paramètres du service.

Tableau 55 – PPM Error

Nom de paramètre	ind
Argument	M
CREP	M
No Data Send	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

No Data Send

Ce paramètre indique l'échec de l'émission de données.

6.3.4.3.7 CPM Get Cons Data

Ce service local est utilisé pour lire les données du tampon de réception d'un consommateur. Le Tableau 56 montre les paramètres du service.

Tableau 56 – Get Cons Data

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M
Data (Données)		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Data

Ce paramètre contient les données de la mémoire tampon de réception.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.8 CPM Get cons status

Ce service local est utilisé pour lire le statut d'APDU du tampon de réception d'un consommateur. Le Tableau 57 montre les paramètres du service.

Tableau 57 – CPM Get cons status

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M
Status		M
Recv Counter		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Status

Ce paramètre contient l'état de l'APDU.

Recv Counter

Ce paramètre contient le compteur des réceptions.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.9 CPM Set RedRole

Ce service local est utilisé pour établir le statut de redondance de la CR. Le Tableau 58 montre les paramètres du service.

Tableau 58 – CPM Set RedRole

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Red Role	M	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Red Role

Ce paramètre détermine le rôle de la redondance (primaire, secours) de la CR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.10 CPM Activate

Ce service local est utilisé pour activer la le diagramme de protocole du consommateur (CPM). Le Tableau 59 montre les paramètres du service.

Tableau 59 – CPM Activate

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
Rx Option	M	

Nom de paramètre	req	cnf
RT Class	M	
Startup Mode	C	
Media Redundancy	C	
Redundant Frame ID	C	
Redundant Watchdog Timer	C	
DFP	C	
Sub Frame Position	C	
Expected Length	M	
Start Ton	M	
Datahold Factor	M	
Default Value	M	
Default Status	M	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

DA

Ce paramètre doit être utilisé pour l'adresse MAC de la destination.

SA

Ce paramètre doit être utilisé pour l'adresse MAC de la source.

Frame ID

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE CR.

Rx Option

Cet attribut conditionnel spécifie les options de réception du client et est constitué des attributs suivants:

RT Class

Cet attribut contient la classe de communication du consommateur.

Startup Mode

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique le mode de démarrage de l'AR correspondante.

Media Redundancy

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'utilisation de MRPD.

Redundant Frame ID

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient l'identificateur de la trame redondante.

Redundancy Watchdog Timer

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "Media Redundancy" est mise à "TRUE". Il contient la valeur du temporisateur chien de garde de redondance (redundancy watchdog timer).

DFP

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "RT Class" est "RT_CLASS_3". Il indique l'usage de DFP.

Sub Frame Position

Cet attribut conditionnel est présent lorsque la valeur de l'attribut "DFP" est "TRUE". Il définit le numéro de position de la sous-trame.

Expected Length

Ce paramètre contient la longueur prévue pour la CR.

Start Ton

Ce paramètre doit être mis à "true" si la temporisation doit être contrôlée immédiatement. Autrement, elle commence avec la réception de la première trame.

Datahold Factor

Ce paramètre contient le facteur de conservation des données.

Default Value

Ce paramètre contient la valeur initiale des données du tampon de réception.

Default Status

Ce paramètre contient la valeur initiale du statut d'APDU du tampon de réception.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned8

Valeur permise: CTEXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.4.3.11 CPM NoData

Ce service local est utilisé pour indiquer que le diagramme de protocole d'un consommateur (CPM) n'a pas reçu de données dans les limites du temporisateur du Data Hold Time (temps de maintien de données). Le Tableau 60 montre les paramètres du service.

Tableau 60 – CPM NoData

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		M
CREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

6.3.4.3.12 CPM Stop

Ce service local sert à arrêter le diagramme de protocole du consommateur (CPM) et, donc, la réception. Le Tableau 61 montre les paramètres du service.

Tableau 61 – CPM Stop

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		M
CREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre contient l'identificateur local de la CR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

6.3.5 ASE Real time acyclic

6.3.5.1 Vue d'ensemble

Le protocole de temps réel acyclique (real-time acyclic (RTA)) est un concept d'écriture et de lecture de données acycliques au moyen de services acquittés. Il utilise des services IEEE 802.3 ou UDP.

6.3.5.2 Spécification de la classe RTA

6.3.5.2.1 Généralités

L'ASE "RTA" définit pour chaque relation de communication (CR) un type d'objet RTA:

6.3.5.2.2 Modèle

Un objet RTA est décrit par le modèle suivant:

ASE: ASE RTA
CLASS: RTA
CLASS ID: pas utilisé
PARENT CLASS: TOP
ATTRIBUTS:

1	(m) Attribut-clé:	CREP
2	(m) Attribut:	Protocol machine Parameter
2.1	(m) Attribut:	DA
2.2	(m) Attribut:	SA
2.3	(m) Attribut:	Frame ID

2.4	(m) Attribut:	VLAN Prio
2.5	(m) Attribut:	VLAN ID
2.6	(m) Attribut:	RTA Timeout Factor
2.7	(m) Attribut:	M Retry
2.8	(m) Attribut:	Send Seq Num
2.9	(m) Attribut:	Ack Seq Num
2.10	(m) Attribut:	Dst IP
2.11	(m) Attribut:	Src IP
2.12	(m) Attribut:	Transport

SERVICES:

1	(m) OpsService:	APMS Activate
2	(m) OpsService:	APMR Activate
3	(m) OpsService:	APMS A Data
4	(m) OpsService:	APMR A Data
5	(m) OpsService:	APMR Ack
6	(m) OpsService:	APMS Error
7	(m) OpsService:	APMR Error
8	(m) OpsService:	APMS Close
9	(m) OpsService:	APMR Close

6.3.5.2.3 Attributs**CREP**

Cet attribut-clé identifie l'instance du diagramme de protocole auquel les données appartiennent.

Type d'attribut: Unsigned32

Protocol machine Parameter

Cet attribut contient les attributs suivants pour décrire le comportement des diagrammes de protocole:

DA

Cet attribut contient l'adresse physique de l'appareil de destination selon l'IEEE 802.

Type d'attribut: OctetString[6]

SA

Cet attribut contient l'adresse physique de l'appareil selon l'IEEE 802.3.

Type d'attribut: OctetString[6]

Frame ID

Cet attribut contient l'identificateur des données au sein de la trame.

Type d'attribut: Unsigned16

Valeurs permises: ALARM_HIGH, ALARM_LOW

VLAN Prio

Cet attribut contient la priorité de la trame selon l'IEEE 802.1Q.

Type d'attribut: Unsigned8

Valeurs permises: LOW, HIGH

VLAN ID

Cet attribut contient le VLAN ID selon l'IEEE 802.1Q.

Type d'attribut: Unsigned16

Valeur permise: 0

RTA Timeout Factor

Cet attribut contient le facteur de temporisation au sein d'une trame de données RTA envoyée qui doit être acquittée Acknowledged (ACK).

Type d'attribut: Unsigned16

Base de temps: 100 ms

Valeurs permises: 1 à 100 (obligatoires), 101 à $2^{16}-1$ (facultatives)

M Retry

Cet attribut contient le nombre de répétitions de tentative si aucun acquittement ACK n'a été reçu dans la limite du RTA Timeout Factor. La plage de valeurs pour les répétitions de tentatives va de 3 à 15.

Type d'attribut: Unsigned16

Send Seq Num

Cet attribut contient le Sender Sequence Number (numéro de séquence d'expéditeur).

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF, 0xFFFF, 0xFFFE

Ack Seq Num

Cet attribut contient l'Acknowledge Sequence Number (numéro de séquence d'acquittement).

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF, 0xFFFF, 0xFFFE

Dst IP

Cet attribut contient l'adresse IP de destination si les services UDP doivent être utilisés.

Type d'attribut: Unsigned32

Src IP

Cet attribut contient l'adresse IP locale si les services UDP doivent être utilisés.

Type d'attribut: Unsigned32

Transport

Cet attribut contient l'acheminement demandé.

Type d'attribut: Unsigned8

Valeurs permises: RTC, UDP

6.3.5.3 Spécification des services RTA

6.3.5.3.1 APMS Activate

Ce service local doit être utilisé pour activer l'Acyclic Protocol Machine Sender (APMS). Le Tableau 62 montre les paramètres du service.

Tableau 62 – APMS Activate

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
VLAN Prio	M	
VLAN ID	M	
RTA Timeout Factor	M	
M Retry	M	
Transport	M	
Dst IP	O	
Src IP	O	

Nom de paramètre	req	cnf
Result(+)		S
CREP		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service. Ces paramètres doivent correspondre aux attributs de l'ASE.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

DA

Ce paramètre doit être utilisé pour l'adresse physique de l'appareil de destination selon l'IEEE 802.3.

SA

Ce paramètre doit être utilisé pour l'adresse physique de l'appareil selon l'IEEE 802.3.

Frame ID

Ce paramètre doit être utilisé pour l'ID de frame (Frame ID).

VLAN Prio

Ce paramètre doit contenir la priorité de la frame selon l'IEEE 802.1Q.

VLAN ID

Ce paramètre doit contenir le VLAN ID selon l'IEEE 802.1Q.

RTA Timeout Factor

Ce paramètre doit contenir le facteur de temporisation.

M Retry

Ce paramètre doit contenir le nombre de répétitions de tentative.

Dst IP

Ce paramètre doit contenir l'adresse IP de destination si les services UDP doivent être utilisés. N'est pas utilisé si les services IEEE 802.3 doivent être utilisés.

Src IP

Ce paramètre doit contenir l'adresse IP locale si les services UDP doivent être utilisés. N'est pas utilisé si les services IEEE 802.3 doivent être utilisés.

Transport

Ce paramètre doit contenir l'acheminement demandé.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned16

Valeur permise: CXTX

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.5.3.2 APMR Activate

Ce service local doit être utilisé pour activer l'Acyclic Protocol Machine Receiver (APMR). Le Tableau 63 montre les paramètres du service.

Tableau 63 – APMR Activate

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
DA	M	
SA	M	
Frame ID	M	
VLAN Prio	M	
VLAN ID	M	
Transport	M	
Dst IP	O	
Src IP	O	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
Error Class		M
Error Code		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service. Ces paramètres doivent correspondre aux attributs de l'ASE.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

DA

Ce paramètre doit être utilisé pour l'adresse physique de l'appareil de destination selon l'IEEE 802.3.

SA

Ce paramètre doit être utilisé pour l'adresse physique de l'appareil selon l'IEEE 802.3.

Frame ID

Ce paramètre doit être utilisé pour l'ID de trame (Frame ID).

VLAN Prio

Ce paramètre doit contenir la priorité de la trame selon l'IEEE 802.1Q.

VLAN ID

Ce paramètre doit contenir le VLAN ID selon l'IEEE 802.1Q.

Transport

Ce paramètre doit contenir l'acheminement demandé.

Dst IP

Ce paramètre doit contenir l'adresse IP de destination si les services UDP doivent être utilisés. N'est pas utilisé si les services IEEE 802.3 doivent être utilisés.

Src IP

Ce paramètre doit contenir l'adresse IP locale si les services UDP doivent être utilisés.
N'est pas utilisé si les services IEEE 802.3 doivent être utilisés.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Class

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned16

Valeur permise: CTXT

Error code

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.5.3.3 APMS A Data

Ce service doit être utilisé pour envoyer des données RTA. Après qu'une trame de données RTA (RTA Data frame) a été envoyée, elle doit être acquittée dans un délai spécifié (basé sur le RTA Timeout Factor). Si aucun acquittement (ACK) n'arrive, la trame de données sera envoyée de nouveau un nombre spécifié de fois (M Retry). Si aucun ACK n'arrive du tout, une APMS Error (erreur d'APMS) sera indiquée (TIMEOUT). La communication utilise les numéros de séquence pour s'assurer une séquence de données appropriée. Les deux points d'extrémité de communication doivent être établis pour la communication afin d'assurer l'établissement du numéro approprié de séquence. Un seul service peut être traité à la fois. Par conséquent, le service doit être terminé avant qu'un nouveau ne puisse être émis (le résultat INVALID_STATE doit être retourné s'il n'est pas terminé). Le Tableau 64 montre les paramètres du service.

Tableau 64 – APMS A Data

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Data (Données)	M	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
ERRCLS		M
ERRCODE		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service. Ces paramètres doivent correspondre aux attributs de l'ASE.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

Data

Ce paramètre doit être utilisé pour les données RTA-SDU à envoyer. La longueur est limitée par la taille des ADPU.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

ERRCLS

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned16

Valeur permise: CTXT

ERRCODE

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.5.3.4 APMR A Data

Ce service local peut être utilisé pour recevoir des données RTA. Le Tableau 65 montre les paramètres du service.

Tableau 65 – APMR A Data

Nom de paramètre	ind
Argument	M
CREP	M
Data	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

Data

Ce paramètre contient les données RTA-SDU reçues.

Result (+)

Ce paramètre indique que la demande de service a réussi.

6.3.5.3.5 APMR Ack

Ce service doit être utilisé pour envoyer un RTA ACK après réception de RTA Data (données RTA). Après réception de RTA Data, celles-ci doivent être acquittées en envoyant un ACK. L'envoi de l'ACK signale à l'expéditeur que les données Data ont été reçues avec succès et d'autres données peuvent être envoyées. Si l'ACK n'est pas envoyé, l'expéditeur répète la tentative et temporisera si aucun ACK n'arrive du tout. Un seul appel de service peut être traité à la fois. Par conséquent, le service doit être terminé avant qu'un nouveau ne puisse être émis (le résultat INVALID_STATE est mis s'il n'est pas terminé). Le Tableau 66 montre les paramètres du service.

Tableau 66 – APMR Ack

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M
Result (-)		S
CREP		M
ERRCLS		M
ERRCODE		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

ERRCLS

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned16

Valeur permise: CTEXT

ERRCODE

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: INVALID_STATE

6.3.5.3.6 APMS error

Ce service local peut être utilisé pour indiquer une APMS error (erreur d'APMS). Le Tableau 67 montre les paramètres du service.

Tableau 67 – APMS Error

Nom de paramètre	ind
Argument	M
CREP	M
ERRCLS	M
ERRCODE	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

ERRCLS

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned16

Valeur permise: voir Tableau 68

ERRCODE

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: voir Tableau 68

Tableau 68 – APMS Error ERRCLS/ERRCODE

ERRCLS	ERRCODE	Signification
PROTOCOL	TIMEOUT	Une temporisation d'émission s'est produite.
PROTOCOL	SEQNUM	Discordance de numéro de séquence.
PROTOCOL	LMPM	LMPM error (Erreur de LMPM)

6.3.5.3.7 APMR Error

Ce service local peut être utilisé pour indiquer une APMR error (erreur d'APMR). Le Tableau 69 montre les paramètres du service.

Tableau 69 – APMR Error

Nom de paramètre	ind
Argument	M
CREP	M
ERRCLS	M
ERRCODE	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

ERRCLS

Ce paramètre sélectionne la classe d'erreur.

Type: Unsigned16

Valeur permise: voir Tableau 70

ERRCODE

Ce paramètre sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: voir Tableau 70

Tableau 70 – APMR Error ERRCLS/ERRCODE

ERRCLS	ERRCODE	Signification
PROTOCOL	SEQNUM	Discordance du numéro de séquence des données reçues
PROTOCOL	LMPM	LMPM error (Erreur de LMPM)

6.3.5.3.8 APMs_Close

Ce service local doit être utilisé pour fermer le diagramme de protocole d'expéditeur (APMS). Le Tableau 71 montre les paramètres du service.

Tableau 71 – APMS_Close

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
ErrCode	U	
Result(+)		S
CREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

ErrCode

Ce paramètre facultatif sélectionne le code d'erreur.

Type: Unsigned16

Valeur permise: Voir l'attribut d'ASE correspondant.

Result (+)

Ce paramètre indique que la demande de service a réussi.

6.3.5.3.9 APMR_Close

Ce service local doit être utilisé pour fermer le diagramme de protocole Receiver (APMR). Le Tableau 72 montre les paramètres du service.

Tableau 72 – APMR_Close

Nom de paramètre	req	cnf
Argument	M	
CREP	M	
Result(+)		S
CREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre identifie l'instance du diagramme de protocole.

Result (+)

Ce paramètre indique que la demande de service a réussi.

6.3.6 ASE "Remote procedure call"

6.3.6.1 Vue d'ensemble

Les concepts d'appel de procédure distante (remote procedure call (RPC)) selon la Publication C706 de The Open Group doivent être appliqués. Cet ASE définit l'usage spécial de la Publication C706 du The Open Group comme protocole de transport.

6.3.6.2 Spécification de la classe RPC

6.3.6.2.1 Modèle

Un objet RPC est décrit par le modèle suivant:

ASE:	ASE RPC
CLASS:	RPC
CLASS ID:	pas utilisé
PARENT CLASS:	TOP
ATTRIBUTES and SERVICES	
1	(m) Attribut-clé: Implicit
2	(m) Attribut: Interface EPMMap
2.1	(o) OpsService: Insert
2.2	(o) OpsService: Delete
2.3	(m) OpsService: Lookup
2.4	(o) OpsService: Map
2.5	(m) OpsService: LookupHandleFree
2.6	(o) OpsService: InqObject
2.7	(o) OpsService: MgmtDelete
3	(m) Attribut: Interface IOD, IOC, IOS, IOPParameterServer
3.1	(m) OpsService: Connect
3.2	(m) OpsService: Release
3.3	(m) OpsService: Read
3.4	(m) OpsService: Write
3.5	(m) OpsService: Control

6.3.6.2.2 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet est adressé de façon implicite par le service.

Interface EPMMap

Cet attribut contient l'interface du mappeur de point d'extrémité de l'appareil. Il y a exactement une seule instance de cette interface pour chaque appareil.

Interface IOD, IOC, IO system, IOPParameterServer

Cet attribut contient l'interface E/S de l'appareil. Le type de l'interface détermine la fonctionnalité adressée (à savoir appareil E/S, appareil de commande E/S, superviseur E/S ou un serveur de paramètre E/S). A partir de chaque type, il peut avoir plusieurs instances de chaque interface; pour plus de détails voir l'Annexe A.

6.3.6.2.3 Comportement général des services

A n'importe quel moment, au maximum un seul appel de service en cours est pris en charge entre un client RCP et le serveur RPC; donc, les appels de service successifs doivent être

mis en séquence. Il est permis d'avoir des appels vers différents services en parallèle (par exemple: une écriture et lecture en cours en parallèle).

L'abandon explicite ou implicite de l'appel de procédure distante selon la Publication C706 de The Open Group est mis en œuvre par des moyens locaux.

Le mappeur de point final est mappé au port 0x8892 à l'opposé du mappeur de point d'extrémité tel que spécifié par la publication C706 qui est mappé au port 0x0087 (135). Ainsi, les fonctions de la bibliothèque du mappeur de point d'extrémité (lié au port 135) d'un client RPC standard ne sont pas utilisables; le client doit appeler l'interface du mappeur de point d'extrémité (Lookup par exemple) au port 0x8892 et évaluer sa réponse.

La réponse de recherche (lookup) contient pour une instance d'interface particulière dans ses informations Protocol Tower, le port de destination à utiliser pour la communication vers cette instance.

Si un client fait un appel direct vers une instance d'interface, il doit respecter la redirection de port du mappeur de point d'extrémité RPC. La demande initiale du client à l'interface est dirigée vers le port 0x8892; la réponse du serveur est émise à partir de son port. Ce port source de la réponse doit donc être utilisé comme port de destination dans les futurs appels du client vers l'instance d'interface.

6.3.6.3 Spécifications des services RPC

6.3.6.3.1 Interface EPMMap

6.3.6.3.1.1 Insert

Pour obtenir la définition de ce service facultatif, consulter la Publication C706 du The Open Group.

6.3.6.3.1.2 Delete

Pour obtenir la définition de ce service facultatif, consulter la Publication C706 du The Open Group.

6.3.6.3.1.3 Lookup

Pour obtenir la définition de ce service, consulter la Publication C706 du The Open Group.

6.3.6.3.1.4 Map

Pour obtenir la définition de ce service facultatif, consulter la Publication C706 du The Open Group.

6.3.6.3.1.5 LookupHandleFree

Pour obtenir la définition de ce service, consulter la Publication C706 du The Open Group.

6.3.6.3.1.6 InqObject

Pour obtenir la définition de ce service facultatif, consulter la Publication C706 du The Open Group.

6.3.6.3.1.7 MgmtDelete

Pour obtenir la définition de ce service facultatif, consulter la Publication C706 du The Open Group.

6.3.6.3.2 Interface IOD, IOC, IOS, IOPParameterServer

6.3.6.3.2.1 Connect

Ce service confirmé doit être utilisé pour établir une connexion au moyen d'un Connectionless Remote Procedure Call (CL-RPC). Le Tableau 73 indique les paramètres du service.

Tableau 73 – Connect

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result (-)			S	S(=)
PNIO Status			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Args Max

Ce paramètre doit contenir la taille maximale de tampon disponible pour la réponse.

Type: Unsigned32

Args Len

Ce paramètre doit contenir le nombre effectif d'octets du paramètre Args.

Type: Unsigned32

Args

Ce paramètre doit contenir les données d'utilisateur RPC.

Type: OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi.

PNIO Status

Ce paramètre doit être mis à zéro.

Type: Unsigned32

Result(-)

Ce paramètre indique que la demande de service a échoué.

PNIO Status

Ce paramètre doit être mis en fonction de l'erreur courante et doit être constitué des éléments Error code = "IODConnectRes", Error Decode = "PNIO", Error code 1 et Error code 2. Les éléments Error code 1 et Error code 2 doivent contenir le code d'erreur réel conforme à la définition de protocole.

Type: Unsigned32

6.3.6.3.2.2 Release

Ce service confirmé doit être utilisé pour libérer une connexion au moyen de CL-RPC. Le Tableau 74 montre les paramètres du service.

Tableau 74 – Release

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result (-)			S	S(=)
PNIO Status			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Args Max

Ce paramètre doit contenir la taille maximale de tampon disponible pour la réponse.

Type: Unsigned32

Args Len

Ce paramètre doit contenir le nombre effectif d'octets du paramètre Args.

Type: Unsigned32

Args

Ce paramètre doit contenir les données d'utilisateur RPC.

Type: OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi.

PNIO Status

Ce paramètre doit être mis à zéro.

Type: Unsigned32

Result(-)

Ce paramètre indique que la demande de service a échoué.

PNIO Status

Ce paramètre doit être mis en fonction de l'erreur courante et doit être constitué des éléments Error code = "IODReleaseRes", Error Decode = "PNIO", Error code 1 et Error code 2. Error code 1 et Error code 2 doivent contenir le code d'erreur réel selon la définition du protocole.

Type: Unsigned32

6.3.6.3.2.3 Read

Ce service confirmé doit être utilisé pour acheminer un service de lecture de couche supérieure au moyen de CL-RPC. Le Tableau 75 montre les paramètres du service.

Tableau 75 – Read

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result (-)			S	S(=)
PNIO Status			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Args Max

Ce paramètre doit contenir la taille maximale de tampon disponible pour la réponse.

Type: Unsigned32

Args Len

Ce paramètre doit contenir le nombre effectif d'octets du paramètre Args.

Type: Unsigned32

Args

Ce paramètre doit contenir les données d'utilisateur RPC.

Type: OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi.

PNIO Status

Ce paramètre doit être mis à zéro.

Type: Unsigned32

Result(-)

Ce paramètre indique que la demande de service a échoué.

PNIO Status

Ce paramètre doit être mis en fonction de l'erreur courante et doit être constitué des éléments Error code = "IODReadRes", Error Decode = "PNIORW", Error code 1 et Error code 2. Error code 1 et Error code 2 doivent contenir le code d'erreur réel selon la définition du protocole.

Type: Unsigned32

6.3.6.3.2.4 Write

Ce service confirmé doit être utilisé pour acheminer un service d'écriture de couche supérieure au moyen de CL-RPC. Le Tableau 76 montre les paramètres du service.

Tableau 76 – Write

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result (-)			S	S(=)
PNIO Status			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Args Max

Ce paramètre doit contenir la taille maximale de tampon disponible pour la réponse.

Type: Unsigned32

Args Len

Ce paramètre doit contenir le nombre effectif d'octets du paramètre Args.

Type: Unsigned32

Args

Ce paramètre doit contenir les données d'utilisateur RPC.

Type: OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi.

PNIO Status

Ce paramètre doit être mis à zéro.

Type: Unsigned32

Result(-)

Ce paramètre indique que la demande de service a échoué.

PNIO Status

Ce paramètre doit être mis en fonction de l'erreur courante et doit être constitué des éléments Error code = "IODWriteRes", Error Decode = "PNIORW", Error code 1 et Error code 2. Error code 1 et Error code 2 doivent contenir le code d'erreur réel selon la définition du protocole.

Type: Unsigned32

6.3.6.3.2.5 Contrôle

Ce service confirmé doit être utilisé pour acheminer un service de commande de couche supérieure au moyen de CL-RPC. Le

Tableau 77 montre les paramètres du service.

Tableau 77 – Control

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
Args Max	M	M(=)		
Args Len	M	M(=)		
Args	M	M(=)		
Result(+)			S	S(=)
PNIO Status			M	M(=)
Args Max			M	M(=)
Args			M	M(=)
Result (-)			S	S(=)
PNIO Status			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Args Max

Ce paramètre doit contenir la taille maximale de tampon disponible pour la réponse.

Type: Unsigned32

Args Len

Ce paramètre doit contenir le nombre effectif d'octets du paramètre Args.

Type: Unsigned32

Args

Ce paramètre doit contenir les données d'utilisateur RPC.

Type: OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi.

PNIO Status

Ce paramètre doit être mis à zéro.

Type: Unsigned32

Result(-)

Ce paramètre indique que la demande de service a échoué.

PNIO Status

Ce paramètre doit être mis en fonction de l'erreur courante et doit être constitué des éléments Error code = "IOXControlRes", Error Decode = "PNIO", Error code 1 et Error code 2. Error code 1 et Error code 2 doivent contenir le code d'erreur réel selon la définition du protocole.

Type: Unsigned32

6.3.7 ASE IEEE 802.1AB

6.3.7.1 Vue d'ensemble

Les concepts selon la norme IEEE 802.1AB doivent être appliqués. La présente partie de la spécification définit l'usage spécial de Chassis ID, Port ID, Chassis ID Subtype, Port ID Subtype et Time to live. En outre, les extensions de l'OUI concernant les trames et la Base de données de gestion («Management Information Base» (MIB)) sont définies.

De plus, l'IETF RFC 3490 (IDNA) doit être appliquée pour la syntaxe de contenu de Chassis ID et Port ID.

Si le protocole SNMP (Simple Network Management Protocol « protocole simple de gestion de réseau ») est pris en charge, le système doit se conformer aux spécifications de gestion de LLDP dans la MIB de LLDP de base et doit mettre en œuvre les sections de la MIB de LLDP pour le mode d'exploitation mis en œuvre. Si le protocole SNMP n'est pas pris en charge, le système doit fournir une capacité de stockage et de récupération équivalant à la fonctionnalité pour le mode d'exploitation mis en œuvre.

6.3.7.2 Spécification de la classe IEEE 802.1AB

6.3.7.2.1 Modèle

L'objet IEEE 802.1AB est décrit par le modèle suivant:

ASE:	ASE Link Layer Discovery
CLASS:	IEEE 802.1AB
CLASS ID:	pas utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Attribut-clé: Implicit
2	(m) Attribut: Chassis ID
3	(m) Attribut: Port ID
4	(m) Attribut: Multiple Interface Mode
5	(m) Attribut: LLDP Time To Live
6	(o) Attribut: Port Description
7	(o) Attribut: System Name
8	(o) Attribut: System Description
9	(o) Attribut: System Capabilities
10	(m) Attribut: Management Address
11	(o) Attribut: Object Identifier
12	(o) Attribut: List of Organizationally-specific extensions
12.1	(m) Attribut: Organizationally Unique Identifier
12.2	(c) Contrainte: Organizationally Unique Identifier = 00-80-C2
12.2.1	(m) Attribut: List of IEEE 802.1 Subtypes
12.2.1.1	(m) Attribut: Subtype
12.2.1.2	(c) Contrainte: Subtype = PORT_VLAN_ID
12.2.1.2.1	(m) Attribut: Port VLAN Identifier
12.2.1.3	(c) Contrainte: Subtype = PORT_AND_PROTOCOL_VLAN_ID
12.2.1.3.1	(m) Attribut: Flags
12.2.1.3.2	(m) Attribut: PPVID Reference Number
12.2.1.4	(c) Contrainte: Subtype = VLAN_NAME
12.2.1.4.1	(m) Attribut: VLAN ID
12.2.1.4.2	(m) Attribut: VLAN Name
12.2.1.5	(c) Contrainte: Subtype = PROTOCOL_IDENTITY
12.2.1.5.1	(m) Attribut: List of Protocol Identities

12.2.1.5.1.1	(m) Attribut:	Protocol Identity
12.3	(c) Contrainte:	Organizationally Unique Identifier = 00-12-0F
12.3.1	(m) Attribut:	List of IEEE 802.3 Subtypes
12.3.1.1	(m) Attribut:	Subtype
12.3.1.2	(c) Contrainte:	Subtype = MAC_PHY_CONFIGURATION_STATUS
12.3.1.2.1	(m) Attribut:	Auto Negotiation Support And Status
12.3.1.2.2	(m) Attribut:	Advertised Capability
12.3.1.2.3	(m) Attribut:	Operational MAU type
12.3.1.3	(c) Contrainte:	Subtype = POWER_VIA_MDI
12.3.1.3.1	(m) Attribut:	MDI Power Support
12.3.1.3.2	(m) Attribut:	Power Pair
12.3.1.3.3	(m) Attribut:	Power Class
12.3.1.4	(c) Contrainte:	Subtype = LINK_AGGREGATION
12.3.1.4.1	(m) Attribut:	Link Aggregation Status
12.3.1.4.2	(m) Attribut:	Aggregated Port ID
12.3.1.5	(c) Contrainte:	Subtype = MAXIMUM_FRAME_SIZE
12.3.1.5.1	(m) Attribut:	Maximum Frame Size
12.4	(c) Contrainte:	Organizationally Unique Identifier = 00-0E-CF
12.4.1	(m) Attribut:	List of PNIO Subtypes
12.4.1.1	(m) Attribut:	Subtype
12.4.1.2	(c) Contrainte:	Subtype = LLDP_PNIO_DELAY
12.4.1.2.1	(m) Attribut:	PnioDelay
12.4.1.2.1.1	(m) Attribut:	PortRxDelayLocal
12.4.1.2.1.2	(m) Attribut:	PortRxDelayRemote
12.4.1.2.1.3	(m) Attribut:	PortTxDelayLocal
12.4.1.2.1.4	(m) Attribut:	PortTxDelayRemote
12.4.1.2.1.5	(m) Attribut:	CableDelayLocal
12.4.1.3	(c) Contrainte:	Subtype = LLDP_PNIO_PORTSTATUS
12.4.1.3.1	(m) Attribut:	PnioPortStatus
12.4.1.3.1.1	(m) Attribut:	RTClass2PortStatus
12.4.1.3.1.2	(m) Attribut:	RTClass3PortStatus
12.4.1.3.1.2.1	(m) Attribut:	Status
12.4.1.3.1.2.2	(m) Attribut:	Mode
12.4.1.4	(c) Contrainte:	Subtype = LLDP_PNIO_ALIAS
12.4.1.4.1	(m) Attribut:	PnioAlias
12.4.1.5	(c) Contrainte:	Subtype = LLDP_PNIO_MRP_PORT_STATUS
12.4.1.5.1	(m) Attribut:	PnioMRPPortStatus
12.4.1.5.1.1	(m) Attribut:	MRPDomainUUID
12.4.1.6	(c) Contrainte:	Subtype = LLDP_PNIO_INTERFACE_MAC_ADDRESS
12.4.1.6.1	(m) Attribut:	PnioInterfaceMacAddress
12.4.1.7	(c) Contrainte:	Subtype = LLDP_PNIO_PTCP_STATUS
12.4.1.7.1	(m) Attribut:	PnioPTCPStatus
12.4.1.7.1.1	(m) Attribut:	PnioPTCPMasterSourceAddress
12.4.1.7.1.2	(m) Attribut:	PnioPTCPSubdomainUUID
12.4.1.7.1.3	(m) Attribut:	PnioIRDataUUID
12.4.1.7.1.4	(m) Attribut:	PnioLengthOfPeriod
12.4.1.7.1.4.1	(m) Attribut:	Length
12.4.1.7.1.4.2	(m) Attribut:	Validity
12.4.1.7.1.5	(m) Attribut:	PnioRedPeriodBegin
12.4.1.7.1.6	(m) Attribut:	PnioOrangePeriodBegin
12.4.1.7.1.7	(m) Attribut:	PnioGreenPeriodBegin
12	(m) Attribut:	List of Remote Systems Data
13.1	(m) Attribut:	Port ID

13.2	(m) Attribut:	Chassis ID
13.3	(m) Attribut:	MAC Address
13.4	(m) Attribut:	LLDP Time To Live
13.5	(o) Attribut:	Port Description
13.6	(o) Attribut:	System Name
13.7	(o) Attribut:	System Description
13.8	(o) Attribut:	System Capabilities
13.9	(o) Attribut:	Management Address
13.10	(o) Attribut:	Object Identifier
13.11	(m) Attribut:	Line Delay
13.12	(m) Attribut:	List of Organizationally-specific extensions
13.12.1	(m) Attribut:	Organizationally Unique Identifier
13.12.2	(c) Contrainte:	Organizationally Unique Identifier = 00-80-C2
13.12.2.1	(m) Attribut:	List of IEEE 802.1 Subtypes
13.12.2.1.1	(m) Attribut:	Subtype
13.12.2.1.2	(c) Contrainte:	Subtype = PORT_VLAN_ID
13.12.2.1.2.1	(m) Attribut:	Port VLAN Identifier
13.12.2.1.3	(c) Contrainte:	Subtype = PORT_AND_PROTOCOL_VLAN_ID
13.12.2.1.3.1	(m) Attribut:	Flags
13.12.2.1.3.2	(m) Attribut:	PPVID Reference Number
13.12.2.1.4	(c) Contrainte:	Subtype = VLAN_NAME
13.12.2.1.4.1	(m) Attribut:	VLAN ID
13.12.2.1.4.2	(m) Attribut:	VLAN Name
13.12.2.1.5	(c) Contrainte:	Subtype = PROTOCOL_IDENTITY
13.12.2.1.5.1	(m) Attribut:	List of Protocol Identities
13.12.2.1.5.1.1	(m) Attribut:	Protocol Identity
13.12.3	(c) Contrainte:	Organizationally Unique Identifier = 00-12-0F
13.12.3.1	(m) Attribut:	List of IEEE 802.3 Subtypes
13.12.3.1.1	(m) Attribut:	Subtype
13.12.3.1.2	(c) Contrainte:	Subtype = MAC_PHY_CONFIGURATION_STATUS
13.12.3.1.2.1	(m) Attribut:	Auto Negotiation Support And Status
13.12.3.1.2.2	(m) Attribut:	Advertised Capability
13.12.3.1.2.3	(m) Attribut:	Operational MAU type
13.12.3.1.3	(c) Contrainte:	Subtype = POWER_VIA_MDI
13.12.3.1.3.1	(m) Attribut:	MDI Power Support
13.12.3.1.3.2	(m) Attribut:	Power Pair
13.12.3.1.3.3	(m) Attribut:	Power Class
13.12.3.1.4	(c) Contrainte:	Subtype = LINK_AGGREGATION
13.12.3.1.4.1	(m) Attribut:	Link Aggregation Status
13.12.3.1.4.2	(m) Attribut:	Aggregated Port ID
13.12.3.1.5	(c) Contrainte:	Subtype = MAXIMUM_FRAME_SIZE
13.12.3.1.5.1	(m) Attribut:	Maximum Frame Size
13.12.4	(c) Contrainte:	Organizationally Unique Identifier = 00-0E-CF
13.12.4.1	(m) Attribut:	List of PNIO Subtypes
13.12.4.1.1	(m) Attribut:	Subtype
13.12.4.1.2	(c) Contrainte:	Subtype = LLDP_PNIO_DELAY
13.12.4.1.2.1	(m) Attribut:	PnioDelay
13.12.4.1.2.1.1	(m) Attribut:	PortRxDelayLocal
13.12.4.1.2.1.2	(m) Attribut:	PortRxDelayRemote
13.12.4.1.2.1.3	(m) Attribut:	PortTxDelayLocal
13.12.4.1.2.1.4	(m) Attribut:	PortTxDelayRemote
13.12.4.1.2.1.5	(m) Attribut:	CableDelayLocal
13.12.4.1.3	(c) Contrainte:	Subtype = LLDP_PNIO_PORTSTATUS

13.12.4.1.3.1	(m) Attribut:	PnioPortStatus
13.12.4.1.3.1.1	(m) Attribut:	RTClass2PortStatus
13.12.4.1.3.1.2	(m) Attribut:	RTClass3PortStatus
13.12.4.1.3.1.2.1	(m) Attribut:	Status
13.12.4.1.3.1.2.2	(m) Attribut:	Mode
13.12.4.1.4	(c) Contrainte:	Subtype = LLDP_PNIO_ALIAS
13.12.4.1.4.1	(m) Attribut:	PnioAlias
13.12.4.1.5	(c) Contrainte:	Subtype = LLDP_PNIO_MRP_PORT_STATUS
13.12.4.1.5.1	(m) Attribut:	PnioMRPPortStatus
13.12.4.1.5.1.1	(m) Attribut:	MRPDomainUUID
13.12.4.1.6	(c) Contrainte:	Subtype = LLDP_PNIO_INTERFACE_MAC_ADDRESS
13.12.4.1.6.1	(m) Attribut:	PnioInterfaceMacAddress
13.12.4.1.7	(c) Contrainte:	Subtype = LLDP_PNIO_PTCP_STATUS
13.12.4.1.7.1	(m) Attribut:	PnioPTCPStatus
13.12.4.1.7.1.1	(m) Attribut:	PnioPTCPMasterSourceAddress
13.12.4.1.7.1.2	(m) Attribut:	PnioPTCPSubdomainUUID
13.12.4.1.7.1.3	(m) Attribut:	PnioIRDataUUID
13.12.4.1.7.1.4	(m) Attribut:	PnioLengthOfPeriod
13.12.4.1.7.1.4.1	(m) Attribut:	Length
13.12.4.1.7.1.4.2	(m) Attribut:	Validity
13.12.4.1.7.1.5	(m) Attribut:	PnioRedPeriodBegin
13.12.4.1.7.1.6	(m) Attribut:	PnioOrangePeriodBegin
13.12.4.1.7.1.7	(m) Attribut:	PnioGreenPeriodBegin

SERVICES:

1 (m) OpsService: Remote Systems Data Change

6.3.7.2.2 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet est adressé de façon implicite par le service.

Multiple Interface Mode

Cet attribut commande le comportement de l'ASE IEEE 802.1AB concernant l'emplacement du nom du port et du NameOfStation.

La valeur FALSE indique que l'attribut "Chassis ID" contient le NameOfStation et l'attribut "PortID" contient le nom du port.

La valeur TRUE indique que l'attribut "Chassis ID" est défini par des moyens locaux et l'attribut "Port ID" contient le nom du port + "." + le NameOfStation.

Type d'attribut: Boolean

Valeur par défaut: FALSE

Chassis ID

Cet attribut non volatil contient le Chassis ID qui doit être utilisé comme sous-type de Chassis ID attribué localement en conformité avec l'IEEE 802.1AB. Ce champ doit contenir une VisibleString indiquant l'identificateur spécifique du châssis particulier dans ce système. La valeur de l'attribut dépend de l'attribut Multiple Interface Mode défini en dessous. Si Multiple Interface Mode est mis à FALSE, cet attribut contient le NameOfStation, autrement le contenu de cet attribut est défini par des moyens locaux.

Type d'attribut: VisibleString[240]

Port ID

Cet attribut contient le nom du port local qui doit être utilisé comme sous-type de Port ID attribué localement en conformité avec l'IEEE 802.1AB.

Si l'attribut Multiple Interface Mode est mis à FALSE, l'attribut contient le nom du port sous la forme « port-xyz » ou « port-xyz-rstuv » où x, y, z sont compris dans la plage «0» à «9» de 001 à 255 maximum et r, s, t, u, v sont compris dans la plage «0» à «9» de 00 000 à 65 535 maximum. Les valeurs "port-001" ou "port-001-00000" doivent être utilisées pour le premier sous-module de port pour une interface dans la baie 0. Les valeurs x, y, z doivent être utilisées pour identifier le numéro de port. Les valeurs r, s, t, u, v doivent être utilisées pour identifier la baie.

Si l'attribut Multiple Interface Mode est mis à TRUE, l'attribut Own Port ID contient le nom du port (« port-xyz » ou « port-xyz-rstuv ») + "." + le NameOfStation.

Type d'attribut: VisibleString[240]

NOTE 1 Dans le contexte du protocole LLDP, un DTE (équipement terminal de traitement de données (ETTD)) est appelé un port. Il ne signifie pas un port selon le contexte UDP.

LLDP Time To Live

Cet attribut indique le nombre de secondes pendant lesquelles l'agent LLDP destinataire doit considérer comme valides les informations associées à cet identificateur de châssis et de port. Il doit être conforme à l'IEEE 802.1AB.

Type d'attribut: Unsigned16

Valeurs permises: 20

Port Description

Cet attribut contient une chaîne alphanumérique qui indique la description de port. Il doit être conforme à l'IEEE 802.1AB.

Si l'IETF RFC 2863 est mise en œuvre, l'utilisation de l'objet ifDescr est recommandée pour ce champ.

System Name

Cet attribut contient une chaîne alphanumérique qui indique le nom attribué administrativement du système. Il doit être conforme à l'IEEE 802.1AB.

Il convient que ce soit le nom de domaine pleinement qualifié du système. Si des mises en œuvre prennent en charge l'IETF RFC 3418, l'utilisation de l'objet sysName est recommandée pour ce champ.

System Description

Cet attribut contient une chaîne alphanumérique qui est la description textuelle de l'entité de réseau. Il doit être conforme à l'IEEE 802.1AB.

Il convient que cette valeur inclue le nom complet et l'identification de version du type de matériel, de système d'exploitation logiciel et du logiciel de mise en réseau relatifs au système. Si des mises en œuvre prennent en charge l'IETF RFC 3418, l'utilisation de l'objet sysDescr est recommandée pour ce champ.

System Capabilities

Cet attribut contient une carte de bits des capacités qui définissent la/les fonction(s) primaire(s) du système. Les positions de bits pour chaque fonction et la norme ou base de données MIB associée qui sont susceptibles - mais cela n'est pas garanti - d'être prises en charge sont énumérées dans le Tableau 78. Un 1 (un) binaire dans le bit associé indique l'existence de la capacité en question. Il doit être conforme à l'IEEE 802.1AB.

NOTE 2 Des systèmes individuels peuvent indiquer plus d'une capacité fonctionnelle mise en œuvre (par exemple: une capacité de pont et une capacité de routeur).

Tableau 78 – System capabilities

Position	Signification

Position	Signification
Bit 0	Autre
Bit 1	Répétiteur
Bit 2	Pont
Bit 3	Point d'accès au réseau local sans fil (WLAN)
Bit 4	Routeur
Bit 5	Téléphone
Bit 6	Appareil câblé selon les DOC SIS (spécifications d'interface du service de transmission de données par câble)
Bit 7	Station seulement

Management Address

Cet attribut contient un OctetString indiquant l'adresse de gestion particulière. Il doit être conforme à l'IEEE 802.1AB.

Il convient que l'adresse retournée soit la plus appropriée pour l'utilisation en gestion, typiquement une adresse de couche 3 telle que l'adresse IPv4 "192.168.254.10".

Object Identifier

Cet attribut identifie le type de composant matériel ou d'entité protocole associé(e) à l'adresse de gestion indiquée. Si aucun identificateur d'objet n'est disponible, ce champ ne doit pas être fourni. Il doit être conforme à l'IEEE 802.1AB.

NOTE 3 Le numéro d'interface et l'identificateur d'objet sont inclus pour aider à la découverte de NMS (Network Management System «système de gestion réseau») en indiquant un point de départ spécifique à une entreprise ou autres points de départ pour la recherche, tels que l'Interface (voir IETF RFC 2863) ou la base Entity MIB (voir IETF RFC 2737).

List of Organizationally-specific extensions

Cette liste d'attributs contient les attributs suivants:

Organizationally Unique Identifier

Cet attribut contient un OUI (Identificateur propre à une organisation) qui est fourni pour permettre à des organisations différentes, telles que l'IEEE 802, l'IETF, ainsi qu'à des vendeurs individuels d'équipements et de logiciels, de définir des données de Longueur/Type/Valeur (Type Length Values (TLV)) qui annoncent des informations à des entités distantes.

List of IEEE 802.1 Subtypes

Cette liste d'attributs contient les attributs suivants:

Subtype

Cet attribut doit être conforme à l'IEEE 802.1AB.

Valeurs permises:
PORT_AND_PROTOCOL_VLAN_ID, VLAN_NAME,
PROTOCOL_IDENTITY

PORT_VLAN_ID,

Port VLAN Identifier

Cet attribut contient le VLAN ID pour le port de pont tel que défini dans l'IEEE 802.1Q. Une valeur de zéro doit être utilisée si le système ne connaît pas l'identificateur VLAN du port ou ne prend pas en charge le fonctionnement VLAN par port.

Flags

Cet attribut contient une carte de bits indiquant la capacité et le statut des VLAN par port et par protocole.

Valeurs permises: PPVID_SUPPORTED, PPVID_ENABLED

PPVID Reference Number

Cet attribut contient le numéro de référence des VALN ID par port et par protocole pour cette station 802 LAN.

Si le port n'est pas capable de prendre en charge des VLAN par port et par protocole et/ou le port n'est activé avec aucun VLAN par protocole par port, il convient que le numéro PPVID soit zéro.

VLAN ID

Cet attribut contient le numéro de VID associé au nom de VLAN.

VLAN Name

Cet attribut contient le nom du VLAN.

Type d'attribut: OctetString[32]

Si des mises en œuvre prennent en charge l'IETF RFC 2674, l'utilisation de l'objet dot1QVLANStaticName est recommandée pour ce champ.

List of Protocol Identities

Cette liste d'attributs contient les attributs suivants:

Protocol Identity

Cet attribut contient les n premiers octets du protocole après les adresses de couche 2 (à savoir débutant par le champ length/type) que l'expéditeur souhaiterait annoncer. La valeur de n est déterminée par la nécessité pour le protocole de se désambigüiser lui-même. La chaîne d'informations de protocole doit inclure suffisamment d'octets pour permettre au destinataire d'identifier correctement le protocole et sa version.

NOTE 4 Les protocoles « Spanning tree » (Protocole d'interconnexion arborescente), par exemple, auraient besoin d'inclure cinq octets: l'adresse LLC (deux octets), l'ID de protocole (deux octets) plus la version du protocole (un octet).

List of IEEE 802.3 Subtypes

Cette liste d'attributs contient les attributs suivants:

Subtype

Cet attribut doit être conforme à l'IEEE 802.1AB.

Valeurs permises: MAC_PHY_CONFIGURATION_STATUS, POWER_VIA_MDI, LINK_AGGREGATION, MAXIMUM_FRAME_SIZE

Auto Negotiation Support And Status

Cet attribut contient une carte de bits qui identifie la prise en charge et le statut actuel d'auto-négociation de la station LAN IEEE 802.3 locale tels que définis dans le Tableau 79. Si le bit Autonegotiation Support de prise en charge de l'autonégociation, (bit 0), a la valeur un (1) et le bit Autonegotiation Status de statut d'autonégociation, (bit 1), a la valeur zéro (0), le mode opérationnel de sous-couches dépendant des supports physiques IEEE 802.3 sera déterminé par la valeur par défaut du champ configuration plutôt que par l'autonégociation.

Tableau 79 – Auto negotiation support and status

Position	Signification
Bit 0	Prise en charge de l'autonégociation (0 – non prise en charge, 1 – prise en charge)
Bit 1	Statut d'autonégociation (0 – désactivée, 1 – activée)

NOTE 5 Cet attribut peut être écrit au moyen du service «Write PD Port Data Adjust».

Advertised Capability

Cet attribut contient une valeur entière telle que définie par l'objet ifMauAutoNegCapAdvertisedBits dans l'IETF RFC 4836.

Operational MAU type

Cet attribut est une valeur entière indiquant le type MAU (Medium Attachment Unit «unité de raccordement au support») opérationnel de l'appareil expéditeur. Cette valeur est dérivée de la position dans la liste du dot3MauType correspondant tel qu'énuméré dans l'IETF RFC 4836 (ou révisions ultérieures) et est égale au dernier nombre dans le dot3MauType correspondant. Pour les types MAU qui ne sont pas énumérés dans l'IETF RFC 4836 (ou révisions ultérieures), la valeur de champ doit être mise à zéro.

MDI Power Support

Cet attribut contient une carte de bits des capacités et du statut d'alimentation de MDI tels que définis dans le Tableau 80.

Tableau 80 – MDI Power Support

Position	Signification
Bit 0	Classe de port (0 – équipement source d'alimentation électrique, 1 – Appareil alimenté en énergie)
Bit 1	Prise en charge de l'alimentation de MDI (0 – non prise en charge, 1 – prise en charge)
Bit 2	Etat de l'alimentation de MDI (0 – désactivée, 1 – activée)
Bit 3	Aptitude de contrôle de paires (0 - la sélection de paires ne peut pas être contrôlée, 1 - la sélection de paires peut être contrôlée)

Power Pair

Cet attribut contient une valeur entière telle que définie par l'objet pethPsePortPowerPairs dans l'IETF RFC 3621.

Power Class

Cet attribut contient une valeur entière telle que définie par l'objet pethPsePortPowerClassification dans l'IETF RFC 3621.

Link Aggregation Status

Cet attribut contient une carte de bits des capacités d'agrégation de liaisons et du statut d'agrégation courant de la liaison tels que définis dans le Tableau 81.

Tableau 81 – Link aggregation status

Position	Signification
Bit 0	Capacité d'agrégation (0 – non prise en charge, 1 – prise en charge)
Bit 1	Statut d'agrégation (0 – désactivée, 1 – activée)

Aggregated Port ID

Cet attribut contient l'identificateur de port agrégé ISO/CEI 15802-1. Un AggPortID est dérivé de l'ifNumber dans l'ifIndex pour l'interface.

Maximum Frame Size

Cet attribut contient une valeur entière indiquant que la taille maximale de trames prises en charge doit être mise à 1 522.

List of PNIO Subtypes

Cette liste d'attributs contient les attributs suivants:

Subtype

Cet attribut contient les valeurs suivantes.

Valeurs permises: LLDP_PNIO_DELAY, LLDP_PNIO_PORTSTATUS, LLDP_PNIO_ALIAS, LLDP_PNIO_MRP_PORT_STATUS, LLDP_PNIO_INTERFACE_MAC_ADDRESS, LLDP_PNIO_PTCP_STATUS

PnioDelay

Cet attribut contient les attributs suivants:

PortRxDelayLocal

Cet attribut contient le retard local de réception du port en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFF

PortRxDelayRemote

Cet attribut contient le retard local de réception du port homologue en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFF

PortTxDelayLocal

Cet attribut contient le retard local d'émission du port en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFF

PortTxDelayRemote

Cet attribut contient le retard local d'émission du port homologue en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFF

CableDelayLocal

Cet attribut contient le retard local de propagation par câble, en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFF

PnioPortStatus

Cet attribut contient les attributs suivants:

RTClass3PortStatus

Cet attribut contient les attributs suivants.

Status

Cet attribut contient le statut de port pour RT_CLASS_3.

Valeurs permises: NOT_USED, IRDATA_CONFIGURED, RTACCLASS3_NEXT_ACTIVE, RTACCLASS3_NEXT_LOADED, RED_ACTIVE

Mode

Cet attribut contient le mode de port pour RT_CLASS_3.

Valeurs permises: STANDARD, OPTIMISED

PnioAlias

Cet attribut contient la VisibleString suivante:

"Valeur de l'attribut Port ID" + "." + "Valeur de Chassis ID"

Type d'attribut: VisibleString

PnioMRPPortStatus

Cet attribut doit être constitué des attributs suivants.

MRPDomainUUID

Cet attribut contient l'UUID de domaine MRP.

Type d'attribut: UUID

PnioInterfaceMacAddress

Cet attribut contient l'adresse MAC selon l'IEEE 802.

Type d'attribut: OctetString[6]

PnioPTCPStatus

Cet attribut doit être constitué des attributs suivants.

PnioPTCPMasterSourceAddress

Cet attribut contient l'adresse MAC selon l'IEEE 802 du maître PTCP.

Type d'attribut: OctetString[6]

PnioPTCPSubdomainUUID

Cet attribut contient l'UUID du sous-domaine PTCP.

Type d'attribut: UUID

PnioIRDataUUID

Cet attribut contient l'UUID des données d'Isochronous Relay (relais isochrone (IR)).

Type d'attribut: UUID

PnioLengthOfPeriod

Cet attribut doit être constitué des attributs suivants.

Length

Cet attribut contient la valeur pour la durée du cycle en nanosecondes. La valeur doit être un multiple de 31 250 nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 0x00007A12 à 0x003D0900 si valide, 0 si non valide

Validity

Cet attribut contient la validité de la longueur.

Valeurs permises: INVALID, VALID

PnioRedPeriodBegin

Cet attribut doit être constitué des attributs suivants.

Offset

Cet attribut contient la valeur du décalage relatif au début du cycle en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000000 à 0x003D08FF si valide, 0 si non valide

Validity

Cet attribut contient la validité du décalage.

Valeurs permises: INVALID, VALID

PnioOrangePeriodBegin

Cet attribut doit être constitué des attributs suivants.

Offset

Cet attribut contient la valeur du décalage relatif au début du cycle en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000000 à 0x003D08FF si valide, 0 si non valide; cette norme exige la valeur 0.

Validity

Cet attribut contient la validité du décalage.

Valeurs permises: INVALID, VALID; cette norme exige la valeur VALID.

PnioGreenPeriodBegin

Cet attribut doit être constitué des attributs suivants.

Offset

Cet attribut contient la valeur du décalage relatif au début du cycle en nanosecondes.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000000 à 0x003D08FF si valide, 0 si non valide

Validity

Cet attribut contient la validité du décalage.

Valeurs permises: INVALID, VALID

List of Remote Systems Data

Cette liste d'attributs contient les mêmes attributs que ceux spécifiés pour Real List of Ports, y compris tous les attributs consécutifs jusqu'à Maximum Frame Size. En outre, les attributs complémentaires suivants sont inclus.

Port ID

Cet attribut a la même définition que l'attribut Own Port ID.

Chassis ID

Cet attribut a la même définition que l'attribut Interface Name.

MAC Address

Cet attribut contient l'adresse physique des appareils conformément à l'adresse MAC de l'IEEE 802.3.

Type d'attribut: OctetString[6]

Line Delay

Cet attribut contient la valeur mesurée du retard de ligne. La base de temps doit être de 1 ns.

Type d'attribut: Unsigned32

6.3.7.2.3 Invocation de l'objet IEEE 802.1AB

Pour l'invocation de l'objet IEEE 802.1AB, les règles suivantes s'appliquent:

- Des objets IEEE 802.1AB doivent exister pour chaque interface.

6.3.7.3 Spécification des services de l'IEEE 802.1AB

Le protocole selon l'IEEE 802.1AB doit être utilisé pour envoyer et recevoir les valeurs d'attributs de l'ASE. L'IEEE 802.1AB utilise le service du LMPM pour émettre et recevoir des informations à destination et en provenance d'autres agents de LLDP.

6.3.7.3.1 Remote systems data change

Ce service local doit être utilisé pour indiquer un changement de certaines Remote Systems Data (données de systèmes distants). Le Tableau 82 montre les paramètres du service.

Tableau 82 – Remote systems data change

Nom de paramètre	ind
Argument	M
Local Port ID	M
List of Neighbors	U
Remote Chassis ID	U
Remote Port ID	U
Remote Cable Delay	U
Remote RT_CLASS_3 Port Status	U
Remote PTCP Status	U
Remote Line Delay	U
Auto Negotiation Support And Status	U
Advertised Capability	U
Operational MAU type	U

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

Local Port ID

Ce paramètre est l'identificateur local.

List of Neighbors

Cette liste de paramètres est constituée des paramètres suivants:

Remote Chassis ID

Ce paramètre contient la valeur de l'attribut de l'ASE.

Remote Port ID

Ce paramètre contient la valeur de l'attribut de l'ASE.

Remote Cable Delay

Ce paramètre contient la valeur de l'attribut de l'ASE.

Remote RT_CLASS_3 Port Status

Ce paramètre contient la valeur de l'attribut de l'ASE.

Remote PTCP Status

Ce paramètre contient la valeur de l'attribut de l'ASE.

Remote Line Delay

Ce paramètre contient la valeur de la somme des attributs suivants de l'ASE:

CableDelayLocal

PortRxDelayLocal

PortTxDelayRemote

Auto Negotiation Support And Status

Ce paramètre contient la valeur de l'attribut de l'ASE.

Advertised Capability

Ce paramètre contient la valeur de l'attribut de l'ASE.

Operational MAU type

Ce paramètre contient la valeur de l'attribut de l'ASE.

6.3.7.3.2 Placeholder (Réceptacle)

Il s'agit d'un paragraphe vide pour être conforme à la Directive CEI partie 2 et à la structure des spécifications de services.

6.3.8 IEEE 802.1AS ASE

6.3.8.1 Vue d'ensemble

Les concepts selon la norme IEEE 802.1AS doivent être appliqués. Cette partie de la spécification définit l'utilisation de l'IEEE 802.1AS.

Le modèle formel de l'ASE IEEE 802.1AS est présenté par la spécification de la classe IEEE 802.1AS, contenant une description de ses attributs et services.

6.3.8.2 Spécification de la classe IEEE 802.1AS

6.3.8.2.1 Modèle

Un objet IEEE 802.1AS est décrit par le modèle suivant:

ASE: IEEE 802.1AS ASE**CLASS:** IEEE 802.1AS**CLASS ID:** pas utilisé**PARENT CLASS:** TOP**ATTRIBUTES:**

- 1 (m) Attribut-clé: pas utilisé
- 2 (m) Attribut: Role
- 3 (m) Attribut: Time
- 4 (m) Attribut: List of Ports
- 4.1 (m) Attribut: PortID
- 4.2 (m) Attribut: Path Delay Boundary

SERVICES:

- 1 (m) OpsService: Local Get Time
- 2 (m) OpsService: Local Set Time
- 3 (m) OpsService: Local Time State Info

6.3.8.2.2 Attributs

Cet attribut-clé n'est pas utilisé.

Expected Role

Cet attribut identifie le rôle attendu de l'interface à l'égard de la synchronisation temporelle.

Type d'attribut: Unsigned8

Valeurs permises: OFF, MASTER, SLAVE

Time

Cet attribut contient l'heure et l'état.

Type d'attribut: TimeStamp

List of Ports

Cet attribut est constitué des éléments suivants:

PortID

Cet attribut identifie un Port d'un pont.

Type d'attribut: Unsigned8

Path Delay Boundary

Cet attribut doit contenir la valeur ON pour les ports d'extrémité d'un domaine 802.1AS.

Type d'attribut: Unsigned8

Valeurs permises: ON, OFF

6.3.8.3 Spécification des services de l'IEEE 802.1AS

Le protocole selon l'IEEE 802.1AB doit être utilisé pour envoyer et recevoir les valeurs d'attributs de l'ASE. L'IEEE 802.1AB utilise le service du LMPM pour émettre et recevoir des informations à destination et en provenance d'autres agents de LLDP.

6.3.8.3.1 Local Get Time

Ce service local peut être utilisé pour obtenir l'heure locale. Le Tableau 83 montre les paramètres du service.

Tableau 83 – Local Get Time

Nom de paramètre	req	cnf
Argument	M	

Nom de paramètre	req	cnf
Result(+)		S(=)
Time		M(=)
Result(-)		S(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Result(+)

Ce paramètre indique que la demande de service a réussi.

Time

Ce paramètre contient la valeur de l'attribut d'ASE équivalent.

Result(-)

Ce paramètre indique que la demande de service a échoué.

6.3.8.3.2 Local Set Time

Ce service local peut être utilisé pour définir l'heure locale. Le Tableau 84 montre les paramètres du service.

Tableau 84 – Local Set Time

Nom de paramètre	req	cnf
Argument	M	
Time	M	
Result(+)		S(=)
Result(-)		S(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Time

Ce paramètre contient la valeur pour l'attribut d'ASE équivalent.

Result(+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

6.3.8.3.3 Local time changed event

Cet événement est initié chaque fois que l'heure a été définie sur une nouvelle valeur. Le Tableau 85 montre les paramètres du service.

Tableau 85 – Local time changed event

Nom de paramètre	ind

Nom de paramètre	ind
Argument	M
OldTime	M
NewTime	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

OldTime

Ce paramètre indique le point dans le temps dans l'ancienne échelle de temps lorsque la nouvelle heure a été définie.

Type d'attribut: TimeStamp

NewTime

Ce paramètre indique le point dans le temps dans l'ancienne échelle de temps lorsque la nouvelle heure a été définie.

Type d'attribut: TimeStamp

6.3.8.4 Invocation de l'objet IEEE 802.1AS

Pour l'invocation de l'objet IEEE 802.1AS, les règles suivantes s'appliquent:

Des objets IEEE 802.1AS doivent exister pour chaque interface:

- Des objets IEEE 802.1AS doivent exister pour chaque interface.

6.3.9 ASE IEEE 802.1D**6.3.9.1 Vue d'ensemble**

L'ASE IEEE 802.1D contrôle le pontage conformément à l'IEEE 802.1D (ponts MAC) prolongé avec les exigences particulières pour RT_CLASS_3.

Les concepts selon la norme IEEE 802.1D doivent être appliqués. Le présent article définit l'utilisation de l'IEEE 802.1D.

Les mécanismes de pontage de l'IEEE 802.1D doivent être désactivés temporairement dans la phase RED. Dans ce cas, le pontage des trames en dehors de la plage d'ID de trame 3 (voir Tableau 276) doit être réalisé en fonction des attributs définis par cet ASE avec le comportement selon le diagramme de protocole Relay RT_CLASS_3. Les valeurs des attributs définissent une table de transmission spéciale.

6.3.9.2 Spécification de la classe IEEE 802.1D**6.3.9.2.1 Généralités**

L'ASE "IEEE 802.1D" définit un type d'objet IEEE 802.1D:

6.3.9.2.2 Modèle

Un objet IEEE 802.1D est décrit par le modèle suivant:

ASE: ASE IEEE 802.1D

CLASS: IEEE 802.1D

CLASS ID: pas utilisé

PARENT CLASS: top

ATTRIBUTES:

1	(m)	Attribut-clé:	Implicit
2	(m)	Attribut:	PD IR Global Data
2.1	(m)	Attribut:	IR Data UUID
2.2	(m)	Attribut:	Max Bridge Delay
2.3	(m)	Attribut:	Number of Ports
2.4	(m)	Attribut:	List of Port Delays
2.4.1	(m)	Attribut:	Max Port Tx Delay
2.4.2	(m)	Attribut:	Max Port Rx Delay
2.4.3	(m)	Attribut:	Max Line Rx Delay
2.4.4	(m)	Attribut:	Yellow Time
3	(o)	Attribut:	PD IR Frame Data
3.1	(m)	Attribut:	Frame Data Properties
3.2	(o)	Attribut:	List of PD IR Frame Data Elements
3.2.1	(m)	Attribut:	Frame Send Offset
3.2.2	(m)	Attribut:	Data Length
3.2.3	(m)	Attribut:	Reduction Ratio
3.2.4	(m)	Attribut:	Phase
3.2.5	(m)	Attribut:	Frame ID
3.2.6	(m)	Attribut:	Ethertype
3.2.7	(m)	Attribut:	Rx Port
3.2.8	(m)	Attribut:	Frame Details
3.2.8.1	(m)	Attribut:	Sync Frame
3.2.8.2	(m)	Attribut:	Meaning Frame Send Offset
3.2.9	(m)	Attribut:	Tx Port Group
3.2.9.1	(m)	Attribut:	Number Of Tx Port Group Array Elements
3.2.9.2	(m)	Attribut:	Tx Port Group Array
4	(m)	Attribut:	PD IR Begin End Data
4.1	(m)	Attribut:	Red Guard
4.1.1	(m)	Attribut:	Start of Red Frame ID
4.1.2	(m)	Attribut:	End of Red Frame ID
4.2	(m)	Attribut:	Number of Ports
4.3	(m)	Attribut:	List of Ports
4.3.1	(m)	Attribut:	Number of Assignments
4.3.2	(m)	Attribut:	List of Assignments
4.3.2.1	(m)	Attribut:	Tx Begin End Assignment
4.3.2.1.1	(m)	Attribut:	Red Orange Period Begin
4.3.2.1.2	(m)	Attribut:	Orange Period Begin
4.3.2.1.3	(m)	Attribut:	Green Period Begin
4.3.2.2	(m)	Attribut:	Rx Begin End Assignment
4.3.2.2.1	(m)	Attribut:	Red Orange Period Begin
4.3.2.2.2	(m)	Attribut:	Orange Period Begin
4.3.2.2.3	(m)	Attribut:	Green Period Begin
4.3.3	(m)	Attribut:	Number of Phases
4.3.4	(m)	Attribut:	List of Phases
4.3.4.1	(m)	Attribut:	Tx Phase Assignment
4.3.4.1.1	(m)	Attribut:	Assigned Value For Reserved Begin
4.3.4.1.2	(m)	Attribut:	Assigned Value For Orange Begin
4.3.4.1.3	(m)	Attribut:	Assigned Value For Reserved End

4.3.4.2	(m) Attribut:	Rx Phase Assignment
4.3.4.2.1	(m) Attribut:	Assigned Value For Reserved Begin
4.3.4.2.2	(m) Attribut:	Assigned Value For Orange Begin
4.3.4.2.3	(m) Attribut:	Assigned Value For Reserved End
5	(m) Attribut:	List of Ports
5.1	(m) Attribut:	Port
5.1.1	(m) Attribut:	Transmission Period Sending
5.1.2	(m) Attribut:	Transmission Period Receiving
5.1.3	(m) Attribut:	Local Port State
5.1.4	(m) Attribut:	Local RTClass 3 Port State
6	(o) Attribut:	Clock Domain Phase Counter

SERVICES:

1	(m) OpsService:	Port State Change
2	(o) OpsService:	Set Port State
3	(o) OpsService:	Flush filtering data base

6.3.9.2.3 Attributs**Implicit**

L'attribut clé « Implicit » indique que l'objet IEEE 802.1D est adressé de façon implicite par le service.

PD IR Global Data

Cet attribut contient les données globales IR utilisées par le système d'ingénierie comme base de calcul. L'appareil doit ajuster ses caractéristiques physiques afin de correspondre aux données globales. Cet attribut se compose des éléments suivants:

IR Data UUID

Cet attribut contient l'identificateur unique du projet temps réel isochrone qui est configuré par le système d'ingénierie

Type d'attribut: UUID

Max Bridge Delay

Cet attribut indique le délai maximal du pont de l'interface en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000 (« inconnu »), 0x00000001 à 0x3B9AC9FF

Number of Ports

Cet attribut indique le nombre de ports que la List of Port Delays (liste des retards de ports) contient.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000001 à 0x000000FF

List of Port Delays

Cette liste contient les retards d'émission et de réception en un port pour chacun des ports, en commençant par le Port 1 jusqu'au numéro de port indiqué par l'attribut Number of Ports. Un élément de la liste est constitué des éléments suivants:

Max Port Tx Delay

Cet attribut indique le retard maximal d'émission du port en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000 (« inconnu »), 0x00000001 à 0x3B9AC9FF

Max Port Rx Delay

Cet attribut indique le retard maximal de réception du port en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000 (« inconnu »), 0x00000001 à 0x3B9AC9FF

Max Line Rx Delay

Cet attribut indique le retard maximal de réception de la ligne en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000 (« aucun retard, transmettre dès que possible »), 0x00000001 à 0x3B9AC9FF

Yellow Time

Cet attribut indique l'heure Yellow en multiples de 1 ns.

L'heure Yellow définit la durée de la période YELLOW. Elle permet à un pont d'optimiser la transmission de trames appartenant aux protocoles PTCP, MRP, RT et IEEE 802.1AS.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00001AE0 à 0x0001E848

PD IR Frame Data

Cet attribut contient les données globales IR utilisées par le système d'ingénierie comme base de calcul. L'appareil doit ajuster ses caractéristiques physiques afin de correspondre aux données globales. Cet attribut se compose des éléments suivants:

Frame Data Properties

Cet attribut contient les éléments suivants:

Forwarding Mode

Cet attribut décrit les paramètres du mode de transfert utilisé pour les trames RT_CLASS_3.

Type d'attribut: Unsigned8

Les valeurs admissibles sont montrées dans le Tableau 86.

Tableau 86 – Valeurs permises de Forwarding Mode

Valeur (décimale)	Signification
0	Mode absolu Le FrameSendOffset contient le temps d'envoi pour une trame transmise.
1	Mode relatif Le FrameSendOffset doit contenir le temps d'envoi pour une trame transmise ou la valeur de "meilleur effort".

Fast Forwarding Multicast MAC Add

Cet attribut décrit les paramètres de l'adresse de destination utilisée pour les trames RT_CLASS_3.

Type d'attribut: Unsigned8

Les valeurs admissibles sont montrées dans le

Tableau 87.

Tableau 87 – Valeurs permises de Fast Forwarding Multicast MAC Add

Valeur (décimale)	Signification
0	Utiliser l'adresse en monodiffusion de destination d'interface MAC (héritage)
1	Utiliser l'adresse en multidiffusion de destination RT_CLASS_3

Valeur (décimale)	Signification
2	Utiliser la Fast Forwarding Multicast MAC Address (adresse MAC en multidiffusion de transmission rapide)

List of PD IR Frame Data Elements

Cette liste contient, pour toutes les trames dans la plage d'ID de trame 3 (voir Tableau 276) que l'appareil doit alimenter, les informations relatives au chemin (c'est-à-dire les ports sur lesquels une trame particulière doit être transmise) et les informations relatives au temps (c'est-à-dire le moment auquel une trame particulière doit être envoyée). De plus, si le mode de transmission est mis au mode absolu, la liste contient les informations relatives au chemin et au temps pour les trames dans la plage d'ID de trame 3 qui doivent être transmises.

Les informations relatives au chemin sont décrites par l'attribut Tx Port Group, les informations relatives au temps par Frame Send Offset.

Tx Port Group indique sur quels ports une trame particulière doit être programmée.

En général, l'attribut Frame Send Offset est mis à une valeur composée par le système d'ingénierie. Un cas spécial consiste à mettre Frame Send Offset à 0, indiquant que la trame doit être envoyée "le plus rapidement possible"; c'est-à-dire que dans la phase RED, la trame quittera l'appareil avec une légère variation temporelle. Pour une trame alimentée par l'appareil, le Frame Send Offset indiqué ici doit correspondre au Frame Send Offset tel qu'indiqué dans la classe CRL.

Si le mode de transmission mis à «mode relatif», l'appareil agit comme expéditeur de transmission relatif, mais comme les informations relatives au temps, les informations relatives au chemin sont également absentes pour les trames devant être transmises et, donc, ce cas d'utilisation ne peut s'appliquer que pour un appareil à deux ports. Les règles de transmission des trames sont les suivantes:

Si une trame particulière à diffusion unique RT_CLASS_3 est reçue sur un port et si l'appareil n'est pas son point final, la trame est transmise avec un retard constant sur l'autre port.

Si une trame particulière à diffusion unique RT_CLASS_3 est reçue sur un port, la trame est transmise avec un retard constant sur l'autre port; indépendamment du fait que l'appareil soit son point final ou non.

Cette liste se compose des éléments de liste suivants:

Frame Send Offset

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission correspondant lorsque la trame particulière doit être envoyée en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000 à 0x003D08FF

NOTE 1 Le trou prévu minimal entre deux trames hors de la plage d'ID de trame 3 (voir Tableau 276) est de 1 120 ns.

Data Length

Cet attribut contient la longueur hors tout des données émises (CSDU Length ou autrement CSFSDU Length); donc, pour une trame condensée de façon dynamique, il indique la longueur de toutes les sous-trames, y compris la sous-trame que l'appareil E/S apporte comme contribution.

Type d'attribut: Unsigned16

Valeurs permises pour RT_CLASS_3: 40 à 1 440

NOTE 2 La limite inférieure de la longueur C_SDU est dérivée de la taille de trame minimale d'une trame Ethernet (= 64 octets). Après avoir soustrait la taille de l'en-tête (à savoir l'adresse de destination, l'adresse et le type de la source) et la taille du pied (à savoir le CRC), il reste 46 octets. Après avoir soustrait la taille de l'en-tête de trame RT (à savoir le Frame ID) et la taille de pied (à savoir le compteur de cycles, le statut des données et le statut de transfert), la longueur minimale de la C_SDU est de 40 octets.

Reduction Ratio

Cet attribut contient le taux de réduction de la trame par rapport à l'horloge d'envoi.

Type d'attribut: Unsigned16

Les valeurs admissibles sont montrées dans le Tableau 88:

Tableau 88 – Valeurs permises de Reduction Ratio

Valeur (décimale)	Signification
1	Obligatoire
2	Obligatoire
4	Obligatoire
8	Obligatoire
16	Obligatoire
32	Facultatif
64	Facultatif
128	Facultatif
256	Facultatif
512	Facultatif
1 024 – 65 535	Réserve
Autre	Facultatif

Phase

Cet attribut indique la phase où la trame particulière doit être envoyée.

Type d'attribut: Unsigned16

Valeurs permises: 1 jusqu'à la valeur courante de l'attribut Reduction Ratio

Frame ID

Cet attribut contient l'identificateur de la trame particulière qui est configurée par le système d'ingénierie.

Type d'attribut: Unsigned32

Les valeurs permises selon le Tableau 89 doivent être utilisées.

Tableau 89 – Frame ID

Valeur (hexadécimale)	Signification	Utilisation
0x0100 – 0x06FF	Dédiées à RT_CLASS_3 (RED) en monodiffusion et en multidiffusion	RED, non redondant, normal ou DFP
0x0700 – 0x0FFF	Dédiées à RT_CLASS_3 (RED) en monodiffusion et en multidiffusion	RED, redondant, normal ou DFP
Autre	Réservé	Ne doit pas être utilisée pour RT_CLASS_3

Ethertype

Cet attribut indique l'Ethertype à utiliser pour la trame.

Type d'attribut: Unsigned16

Valeurs permises: 0x8892

Rx Port

Cet attribut contient le numéro du port de réception. La valeur LOCAL_INJECTION doit être utilisée pour les trames alimentées par cet appareil. La valeur PORT_1 à PORT_255 doit être utilisée pour les trames devant être transmises.

Type d'attribut: Unsigned8

Valeurs permises: LOCAL_INJECTION, PORT_1 to PORT_255

Frame Details

Cet attribut est constitué des éléments suivants:

Sync Frame

Cet attribut indique si la trame particulière est une trame de synchronisation ou une trame normale.

Type d'attribut: Unsigned8

Valeurs permises: NO_SYNC_FRAME, PRIMARY_SYNC_FRAME, SECONDARY_SYNC_FRAME

Meaning Frame Send Offset

Cet attribut indique la signification détaillée de l'attribut Frame Send Offset.

L'attribut doit être mis à TX_FRAME, lorsque l'attribut Frame Send Offset indique l'heure, quand une trame hors de plage d'ID de trame 3 (voir Tableau 276) doit être émise.

NOTE 3 Une trame est reçue localement, lorsque l'attribut Rx Port est mis à LOCAL_INJECTION.

Type d'attribut: Unsigned8

Valeurs permises: TX_FRAME

Tx Port Group

Cet attribut indique où la trame particulière doit être transmise. Il est constitué des éléments suivants:

Number Of Tx Port Group Array Elements

Ce champ doit compter les entrées de Tx Port Group Array successives.

Type d'attribut: Unsigned8

Valeurs permises: 1, 3, 5, 7, 9, 11, 13, 15, ..., 31, 33.

Tx Port Group Array

L'attribut Tx Port Group Array est une matrice d'octets qui doit contenir au moins un octet et au plus 33 octets appelé(s) "Tx Port Group Array Entry" c'est-à-dire "entrée(s) de Tx Port Group Array". Un Tx Port Group Array Element doit consister en au moins une entrée TxPort et au plus huit entrées TxPort appelées entrée de TxPortGroup 0 à entrée de TxPortGroup 7. Par conséquent, le nombre d'octets de TxPortGroup correspond au nombre de ports sur un appareil et doit être calculé comme suit:

$$N = M_{\text{highest}} \text{ DIV } 8 + 1 \quad (20)$$

où

N est le nombre d'octets TxPortGroup ou le nombre d'éléments de la matrice,
 M_{highest} est le plus grand numéro des TxPorts sur l'appareil (maximum 255).

Les entrées TxPort non utilisées doivent être mises à zéro.

Le dernier octet de TxPortGroup (octet N) peut ne pas contenir toutes les 8 entrées TxPort. Si $M_{\text{le plus grand}} \text{ MOD } 8 \neq 7$, l'octet N n'est pas complètement renseigné et les bits restants doivent être mis à zéro et appelés Padding Bits (bits de bourrage).

NOTE 4 Le terme DIV signifie la division sans reste. Le terme MOD signifie le reste de la division.

Bit 0: TxPortEntry_0

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 0$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Tableau 90 – Tx Port Entry

Value (hexadécimale)	Signification
0x00	Emission désactivée
0x01	Emission activée

Le TxPort d'injection locale est toujours placé dans l'entrée TxPortEntry_1 de l'octet de TxPortGroup numéro un. Ainsi, l'injection locale est mappée au port 0.

Bit 1: TxPortEntry_1

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 1$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Bit 2: TxPortEntry_2

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort avec le numéro qui satisfait à $m \text{ MOD } 8 = 2$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Bit 3: TxPortEntry_3

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 3$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Bit 4: TxPortEntry_4

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 4$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Bit 5: TxPortEntry_5

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 5$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Bit 6: TxPortEntry_6

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 6$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

Bit 7: TxPortEntry_7

Ce bit doit être mis avec les valeurs conformes au Tableau 90 si le TxPort ayant le numéro qui satisfait à $m \text{ MOD } 8 = 7$ est présent. Un bit de bourrage doit être utilisé si aucun TxPort n'est présent.

où m est le numéro du TxPort courant avec $1 \leq m \leq N$.

PD IR Begin End Data List

Cette liste contient le Red Guard et les informations de gestion de bande passante pour les classes de trafic RED, ORANGE et GREEN. Il se compose des éléments de liste suivants:

Red Guard

Cet attribut contrôle le comportement des ports de réception sur le diagramme d'états du port de classe 3 en temps réel (Realtime Class 3 Port State Machine (RTC3PSM)) à l'état « Run » (exécuter), à savoir que le port Rx est en mode RED.

Le Red Guard empêche les trames erronées d'entrer dans la classe de trafic RED en tant que première couche de filtrage. Toutes les trames avec un ID de trame inférieur à Start of Red Frame ID ou supérieur à End of Red Frame ID sont bloquées pendant la période RED.

La deuxième couche de filtrage est assurée par les IR Frame Data Elements (éléments de données de trames d'IR). Seule une trame franchissant les deux filtres pénètre dans la classe de trafic RED. Si IR Frame Data est manquant, la seconde couche de filtrage est omise.

Start of Red Frame ID

Cet attribut définit la limite inférieure du Red Guard. La valeur doit être mise à une valeur inférieure au contenu de l'attribut End of Red Frame ID.

Type d'attribut: Unsigned16

Valeurs permises: 0x0080 à 0x7FFF

Valeur par défaut: 0x0100

End of Red Frame ID

Cet attribut définit la limite supérieure du Red Guard.

Type d'attribut: Unsigned16

Valeurs permises: 0x0080 à 0x7FFF

Valeur par défaut: 0x7FFF

Number of Ports

Cet attribut contient le nombre d'éléments dans la List of Ports.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000001 à 0x000000FF

List of Ports

Cette liste contient les informations d'administration de bande passante. Elle est composée des éléments de liste suivants.

Number of Assignments

Cet attribut contient le nombre d'éléments dans la List of Assignments. Un appareil doit prendre en charge au minimum 5 attributions différentes (quatre pour les phases avec des périodes réservées (RED) et une pour les phases sans aucune période réservée).

Dans les attributions avec un minimum jusqu'à deux valeurs différentes Red Orange Period Begin et Orange Period Begin (l'une d'elles pour les phases sans aucune période réservée) et jusqu'à cinq valeurs différentes Green Period Begin (l'une d'elles pour les phases sans aucune période réservée) doivent être prises en charge.

Une phase sans aucune période réservée est indiquée par le fait d'avoir Red Orange Period Begin, Orange Period Begin et Green Period Begin mis à 0x0.

Le nombre d'attributions prises en charge par l'appareil est indiqué par la GSD.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000005 à 0x00000010

List of Assignments

Cette liste contient les affectations. Un élément particulier de la liste contient des informations relatives à la bande passante. Un élément de liste est constitué des attributs suivants:

Tx Begin End Assignment

Cet attribut définit les débuts des périodes RED, ORANGE et GREEN.

Red Orange Period Begin

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission où la période RED doit commencer en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000

Orange Period Begin

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission où la période ORANGE doit commencer en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: Valeurs permises: doit être défini sur Green Period Begin (signifiant qu'aucune période ORANGE n'est définie)

NOTE 5 Une période orange existe uniquement pour les appareils hérités.

Green Period Begin

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission où la période GREEN doit commencer en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000 à 0x003D08FF

Rx Begin End Assignment

Cet attribut définit les débuts des périodes RED, ORANGE et GREEN

Red Orange Period Begin

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission où la période RED doit commencer en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: 0x00000000

Orange Period Begin

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission où la période ORANGE doit commencer en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: doit être mis à Green Period Begin

NOTE 6 Pour un port Rx, il n'existe pas de période orange.

Green Period Begin

Cet attribut indique le décalage relatif au début du cycle de l'horloge d'émission où la période GREEN doit commencer en multiples de 1 ns.

Type d'attribut: Unsigned32

Base de temps: 1 ns

Valeurs permises: doit être mis à la fin de la période RED.

Number of Phases

Cet attribut contient le nombre d'éléments dans la List of Phases utilisée pour affectations. Jusqu'à 16 différentes affectations de phase sont prises en charge.

Une phase de trame particulière est mappée à la phase d'affectation par the la Formule (21)

$$\text{Assignment Phase} = \text{Frame Phase} \text{ MOD } \text{Number of Phases.} \quad (21)$$

où

<i>Assignment Phase</i>	est la phase d'affectation
<i>Frame Phase</i>	est la phase de trame particulière
<i>Number of Phases</i>	est le nombre d'éléments dans la List of Phases.

Type d'attribut: Unsigned32

Valeurs permises: 1, 2, 4, 8, 16

List of Phases

Cet attribut affecte à une phase particulière les débuts de période appropriés.

Pour l'affectation de débuts particuliers de la phase en question, les règles énoncées dans le Tableau 91 s'appliquent.

Tableau 91 – Dépendances de RedOrangePeriodBegin, OrangePeriodBegin, et GreenPeriodBegin

Valeur (hexadécimale)	Signification
RedOrangePeriodBegin < OrangePeriodBegin = GreenPeriodBegin	Seule une période RED est définie.
RedOrangePeriodBegin = OrangePeriodBegin = GreenPeriodBegin = 0	Aucune période réservée n'est définie.
RedOrangePeriodBegin = OrangePeriodBegin < GreenPeriodBegin	Hérité seulement
RedOrangePeriodBegin < OrangePeriodBegin < GreenPeriodBegin	non utilisé
RedOrangePeriodBegin > OrangePeriodBegin	non utilisé
RedOrangePeriodBegin > GreenPeriodBegin	non utilisé
OrangePeriodBegin > GreenPeriodBegin	non utilisé

Tx Phase Assignment

Cet attribut affecte les valeurs des périodes RED, ORANGE et GREEN sur tous les ports Tx pour une phase particulière.

Assigned Value For Reserved Begin

Cet attribut contient le numéro positionnel d'un élément particulier dans la List of Assignments. Le contenu de l'attribut Red Orange Period Begin dans l'attribut d'élément Tx Begin End Assignment est utilisé comme début de la période RED sur le port Tx.

Type d'attribut: Unsigned8

Valeurs permises: 0 à Number of Assignments - 1

Assigned Value For Orange Begin

Cet attribut contient le numéro positionnel d'un élément particulier dans la List of Assignments. Le contenu de l'attribut Orange Period Begin dans l'attribut d'élément Tx Begin End Assignment est utilisé comme début de la période ORANGE sur le port Tx.

Type d'attribut: Unsigned8

Valeurs permises: 0 à Number of Assignments - 1

Assigned Value For Reserved End

Cet attribut contient le numéro positionnel d'un élément particulier dans la List of Assignments. Le contenu de l'attribut Green Period Begin dans l'attribut d'élément Tx Begin End Assignment est utilisé comme début de la période GREEN sur le port Tx.

Type d'attribut: Unsigned8

Valeurs permises: 0 à Number of Assignments - 1

Rx Phase Assignment

Cet attribut affecte les valeurs des périodes RED, ORANGE et GREEN sur tous les ports Rx pour une phase particulière.

Assigned Value For Reserved Begin

Cet attribut contient le numéro positionnel d'un élément particulier dans la List of Assignments. Le contenu de l'attribut Red Orange Period Begin dans l'attribut d'élément Rx Begin End Assignment est utilisé comme début de la période RED sur le port Rx.

Type d'attribut: Unsigned8

Valeurs permises: 0 à Number of Assignments - 1

Assigned Value For Orange Begin

Cet attribut contient le numéro positionnel d'un élément particulier dans la List of Assignments. Le contenu de l'attribut Orange Period Begin dans l'attribut d'élément Rx Begin End Assignment est utilisé comme début de la période ORANGE sur le port Rx.

Type d'attribut: Unsigned8

Valeurs permises: 0 à Number of Assignments - 1

Assigned Value For Reserved End

Cet attribut contient le numéro positionnel d'un élément particulier dans la List of Assignments. Le contenu de l'attribut Green Period Begin dans l'attribut d'élément Rx Begin End Assignment est utilisé comme début de la période GREEN sur le port Rx.

Type d'attribut: Unsigned8

Valeurs permises: 0 à Number of Assignments - 1

List of Ports

Cet attribut doit être constitué des éléments suivants.

Port

Cet attribut est constitué des éléments suivants:

Transmission Period Sending

Cet attribut contient la valeur de la période d'émission courante du port dans le sens d'envoi. La programmation du processus d'envoi conformément à l'ASE IEEE 802.1Q doit être appliquée.

Les règles d'émission suivantes pour RED doivent être appliquées:

Le processus local de mise en file d'attente doit émettre une trame en fonction de Frame Send Offset et de Tx Port Group Array si le calcul de Phase et de Reduction Ratio s'adapte à Phase Number.

Les règles de transmission suivantes pour RED doivent être appliquées:

Le processus local de mise en file d'attente doit émettre une trame reçue antérieurement en fonction de Frame Send Offset, Data Length, Frame ID et Rx Port. Si le calcul de Phase et Reduction Ratio s'adapte à Phase Number, la trame doit être émise sur tous les ports stockés dans Tx Port Group Array.

Dans la limite de la période YELLOW, les règles de transmission de l'IEEE 802.1D doivent être omises.

Les règles d'émission suivantes doivent être appliquées:

Le processus local de mise en file d'attente doit émettre une trame selon que sa taille est plus courte que l'intervalle YELLOW restant.

Les règles de transmission suivantes doivent être appliquées:

Le processus local de mise en file d'attente doit émettre une trame reçue antérieurement si sa taille est plus courte que l'intervalle YELLOW restant.

Dans les limites de la période ORANGE et GREEN, l'IEEE 802.1D doit être appliquée.

La période d'émission doit être commandée par une horloge locale. Cette horloge doit être synchronisée sur le réseau en utilisant des périodes GREEN. La durée de toutes les périodes ne doit pas dépasser la valeur du temps calculée au moyen de l'attribut Send Clock Factor, qui doit être défini par un autre ASE. L'exactitude est déterminée par des classes de conformité.

Type d'attribut: Unsigned8

Valeurs permises: RED (facultative), ORANGE (hérités seulement), GREEN, YELLOW (facultative)

NOTE 7 Comme le montre la Figure 8, l'envoi de trames est divisé en différentes période au niveau du port local. Chaque phase est définie par $T_{Sendclock}$ qui se situe entre 31,25 μ s et 4 ms. Le nombre de trames de chaque période dépend des trames programmées pour la phase courante et peut également être zéro.

Transmission Period Receiving

Cet attribut contient la valeur de la période d'émission courante du port dans le sens de réception.

Dans les limites de la période RED, seules les DLPDU RT_CLASS_3 à temps et les DLPDU de synchronisation PTCP à temps doivent être transmises selon les règles de relais RED. Toutes les autres DLPDU reçues doivent être traitées par le MAC Relay (relais MAC) conformément aux règles de transmission de l'IEEE 802.1D.

Les règles de réception suivantes pour RED doivent être appliquées:

Le processus local de mise en file d'attente doit recevoir une trame en fonction de Frame Send Offset, Data Length, Frame ID et Rx Port et le calcul de Phase et de Reduction Ratio s'adapte à Phase Number.

$$\text{Phase} := (\text{Clock Domain Phase Counter MOD Reduction ratio}) + 1 \quad (22)$$

où

<i>Phase</i>	est la valeur de phase
<i>Clock Domain Phase Counter</i>	est le compteur de phases du domaine d'horloge
<i>Reduction ratio</i>	est le taux de réduction

Les adresses MAC sources IEEE 802.3 reçues ne doivent pas être intégrées dans la table des adresses de transmission afin d'éviter d'influencer l'émission de toutes les autres trames.

Les règles de réception suivantes doivent être appliquées:

Dans les limites de la période ORANGE et GREEN, l'IEEE 802.1D (MAC Relay) doit être appliquée.

La période d'émission doit être commandée par une horloge locale. Cette horloge doit être synchronisée sur le réseau en utilisant des périodes GREEN. La durée de toutes les périodes ne doit pas dépasser la valeur du temps calculée au moyen de l'attribut Send Clock Factor, qui doit être défini par un autre ASE. L'exactitude est déterminée par des classes de conformité.

Type d'attribut: Unsigned8

Valeurs permises: RED (facultative), ORANGE (hérités seulement), GREEN

Local Port State

Cet attribut contient la valeur pour le port selon la gestion de pont IEEE 802.1D.

Type d'attribut: Unsigned8

Valeurs permises: DISCARDING, BLOCKED, FORWARDING

Local RTClass 3 Port State

Cet attribut contient la valeur pour le port en fonction de la phase RED. La valeur OFF signifie "non utilisé" ou "configuré pour la phase RED" ou les préconditions n'ont pas pu être atteintes jusqu'ici. La valeur RTCLASS3_UP signifie que le sens de l'émission est dans la phase RED. La valeur RTCLASS3_RUN signifie que les deux sens sont dans la phase RED.

Type d'attribut: Unsigned8

Valeurs permises: OFF, RTCLASS3_UP, RTCLASS3_RUN

Clock Domain Phase Counter

Cet attribut contient le compte de phases passées. Tous les nœuds utilisés avec RT_CLASS_3 doivent utiliser la même valeur. Elle est obtenue par synchronisation d'horloge PTCP.

Type d'attribut: Unsigned64

6.3.9.3 Spécification des services de l'IEEE 802.1D

6.3.9.3.1 Port state change

Ce service local doit être utilisé pour indiquer un changement de certaines Local Systems Data (données de systèmes locaux). Le Tableau 92 montre les paramètres du service.

Tableau 92 – Port state change

Nom de paramètre	ind
Argument	M
Local Port ID	M
Local Port State	M
Local RTClass 3 Port State	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

Local Port ID

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Port State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local RTClass 3 Port State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

6.3.9.3.2 Set port state

Ce service local doit être utilisé pour établir certaines Local Systems Data (données de systèmes locaux). Le Tableau 93 montre les paramètres du service.

Tableau 93 – Set port state

Nom de paramètre	req
Argument	M
Local Port ID	M
Local Port State	U
Local RTClass 3 Port State	U

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

Local Port ID

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Port State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local RTClass 3 Port State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

6.3.9.3.3 Flush filtering data base

Ce service local doit être utilisé pour vider la base de données de filtrage locale. Le

Tableau 94 montre les paramètres du service.

Tableau 94 – Flush filtering data base

Nom de paramètre	req
Argument	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

6.3.9.4 Invocation de l'objet IEEE 802.1D

Pour l'invocation de l'objet IEEE 802.1D, les règles suivantes s'appliquent:

- Des objets IEEE 802.1D doivent exister pour chaque interface.
- Les services avec le préfixe IFW doivent exister pour chaque interface RT_CLASS_3.

6.3.10 ASE IEEE 802.1Q**6.3.10.1 Vue d'ensemble**

Les concepts selon la norme IEEE 802.1D (Ponts MAC) doivent être appliqués. Cette partie de la spécification définit l'utilisation de la norme IEEE 802.1Q (Réseaux locaux virtuels pontés). Elle définit des extensions pour RT_CLASS_3 concernant le classement par ordre de priorité des trames.

Des processus d'application contiennent des données qui sont émises avec différentes priorités selon l'IEEE 802.1Q. De plus, l'usage de ces priorités doit être étendu dans différents intervalles de temps selon la présente spécification. L'ASE "IEEE 802.1Q" définit des attributs pour déterminer le comportement de temporisation.

Le modèle formel de l'ASE IEEE 802.1Q est présenté par la spécification de la classe IEEE 802.1Q, contenant une description de ses attributs.

6.3.10.2 Spécification de la classe IEEE 802.1Q**6.3.10.2.1 Modèle**

Un objet IEEE 802.1Q est décrit par le modèle suivant:

ASE: ASE IEEE 802.1Q

CLASS: IEEE 802.1Q

CLASS ID: pas utilisé

PARENT CLASS: TOP

ATTRIBUTS:

1	(m) Attribut-clé:	non utilisé
2	(o) Attribut:	Network Control High Queue
3	(m) Attribut:	Network Control Low Queue
4	(m) Attribut:	Prio 7 Send Queue
5	(m) Attribut:	Prio 6
5.1	(o) Attribut:	Red Send Queue
5.2	(m) Attribut:	Prio 6 Send Queue
6	(m) Attribut:	Prio 5 Send Queue
7	(m) Attribut:	Prio 4 Send Queue
8	(m) Attribut:	Prio 3 Send Queue
9	(m) Attribut:	Prio 2 Send Queue
10	(m) Attribut:	Prio 1 Send Queue
11	(m) Attribut:	Prio 0 Send Queue

12	(m) Attribut:	Transmission Period
13	(o) Attribut:	Clock Domain Phase Counter

6.3.10.2.2 Attributs

Cet attribut-clé n'est pas utilisé.

Network Control High Queue

Cet attribut facultatif contient la liste des DLPDU pour synchronisation pour les DLPDU qui ne sont pas planifiées dans les registres de données IR (IR Data records) ou celles qui sont reçues hors du temps planifié. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Network Control Low Queue

Cet attribut contient la liste des DLPDU pour la priorité organisationnelle selon l'IEEE 802.1D pour des protocoles de gestion tels que LLDP, la mesure de retard de ligne PTCP, MRP, et autres. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 7 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 7 de l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 6

Cet attribut est constitué des attributs suivants:

Red Send Queue

Cet attribut facultatif contient la liste des DLPDU pour RT_CLASS_3 et pour les DLPDU pour synchronisation qui avaient été planifiées dans un registre de données IR et reçues à temps. La séquence des trames et l'heure exacte d'envoi doivent être déterminées par d'autres attributs de l'ASE. Après émission, la DLPDU doit être retirée de la file d'attente. Avec le début d'une nouvelle phase la Red Send Queue (file d'attente d'envoi rouge) doit être remplies avec les DLPDU pour la phase courante.

Type d'attribut: OctetString

NOTE 1 La couche application pour périphérie décentralisée définit une List of IR Frame Data Elements (liste d'éléments de données de trames IR) pour déterminer la séquence et l'heure exacte d'envoi des trames.

Prio 6 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 6 selon l'IEEE 802.1Q, y compris les DLPDU pour RT_CLASS_1 et RT_CLASS_UDP. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 5 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 5 selon l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 4 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 4 selon l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 3 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 3 selon l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 2 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 2 selon l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 1 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 1 selon l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Prio 0 Send Queue

Cet attribut contient la liste des DLPDU pour la priorité 0 selon l'IEEE 802.1Q. Après émission, la DLPDU doit être retirée de la file d'attente.

Type d'attribut: OctetString

Transmission Period

Cet attribut contient la valeur de la période d'émission courante du port. Il vise à commander l'utilisation de la largeur de bande au sein du réseau.

NOTE 2 En théorie, chaque appareil pourrait utiliser toujours la vitesse sur câble pour envoyer des trames. Un tel comportement appliqué pendant un certain temps mettrait le système hors service. Par conséquent, il est fourni un mécanisme raisonnable pour limiter les performances d'envoi de chaque appareil.

Dans les limites de la période RED, seules des DLPDU issues de la file d'attente Red Send Queue doivent être émises.

Dans la période GREEN, les DLPDU provenant de toutes les files d'attente doivent être émises selon leurs priorités.

Dans la période GREEN, les règles suivantes doivent être appliquées pour la programmation locale:

- Le processus local de mise en file d'attente doit abandonner toutes les trames RT_CLASS_1 pour la prochaine phase si au début d'une nouvelle période GREEN, la file d'attente Prio 6 Send Queue n'est pas vide. Chaque abandon doit être signalé à l'application locale.
- En moyen, pas plus de 90 pour cent de la largeur de bande ne doit être utilisé pour éviter la surcharge du réseau.

Dans les limites de la période YELLOW, les DLPDU provenant de toutes les files d'attente doivent être émises selon leurs priorités si l'émission de la trame se termine dans les limites de la période. Si la valeur calculée du temps d'émission pour une trame dépasse la période, la première trame de priorité inférieure peut être soumise à essai en option.

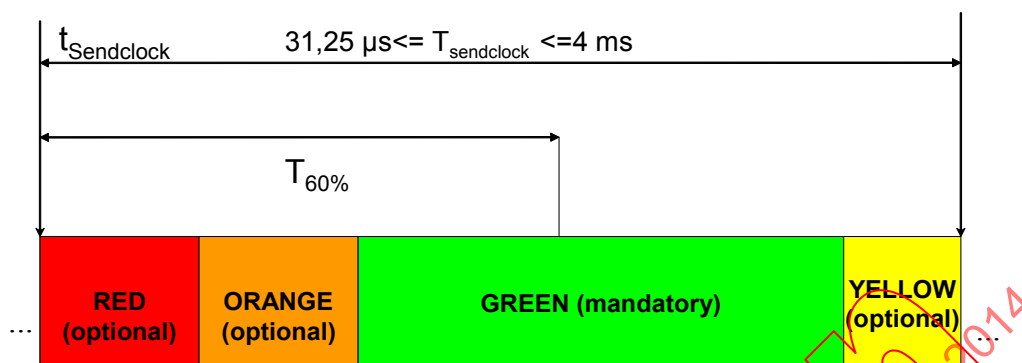
La séquence des périodes doit être RED, GREEN et YELLOW. Il est permis d'omettre la période RED. La période YELLOW doit exister si RED existe) et doit être omise dans le cas contraire.

La période d'émission doit être commandée par une horloge locale. Cette horloge doit être synchronisée sur le réseau en utilisant des périodes GREEN. La durée de toutes les périodes ne doit pas dépasser la valeur du temps calculée au moyen de l'attribut Send Clock Factor, qui doit être défini par un autre ASE. L'exactitude est déterminée par des classes de conformité.

Type d'attribut: Unsigned8

Valeurs permises: RED (facultative), GREEN, YELLOW (facultative)

NOTE 3 Comme le montre la Figure 8, l'envoi de trames est divisé en différentes période au niveau du port local. Chaque phase est définie par $T_{\text{Sendclock}}$ qui se situe entre $31,25 \mu\text{s}$ et 4 ms . Le nombre de trames de chaque période dépend des trames programmées pour la phase courante et peut également être zéro.



Légende

Anglais	Français
$t_{\text{sendclock}}$	$t_{\text{horloge d'envoi}}$
RED (optional)	ROUGE (facultatif)
ORANGE (optional)	ORANGE (facultatif)
GREEN (mandatory)	VERT (obligatoire)
YELLOW (optional).	JAUNE (facultatif)

Figure 8 – Exemple de périodes en un port local

La première et la dernière invocation de la période RED doivent être synchronisées entre le port local et le port de voisinage au moyen du service Remote Systems Data Change (Remote RT_CLASS_3 Port Status).

Clock Domain Phase Counter

Cet attribut contient le compte de phases passées. Tous les nœuds utilisés avec RT_CLASS_3 doivent utiliser la même valeur. Elle est obtenue par synchronisation d'horloge PTCP.

Type d'attribut: Unsigned64

6.3.10.2.3 Invocation de l'objet IEEE 802.1Q

Pour l'invocation de l'objet IEEE 802.1Q, les règles suivantes s'appliquent:

- Des objets IEEE 802.1Q doivent exister pour chaque port.

6.3.10.3 Spécification des services de l'IEEE 802.1Q

Il n'y a pas de services spécifiés.

6.3.11 ASE IEEE 802.3

6.3.11.1 Vue d'ensemble

L'ASE IEEE 802.3 commande l'accès au support.

Les concepts conformes à la norme IEEE 802.3 (Carrier Sense Multiple Access avec méthode d'accès avec détection de collision (CSMA/CD) et spécifications pour la couche physique) doivent être appliqués. Le présent article définit l'utilisation de l'IEEE 802.3.

6.3.11.2 Spécification de la classe IEEE 802.3

6.3.11.2.1 Généralités

L'ASE IEEE 802.3 définit un type d'objet IEEE 802.3:

6.3.11.2.2 Modèle

Un objet IEEE 802.3 est décrit par le modèle suivant:

ASE: ASE IEEE 802.3

CLASS: IEEE 802.3

CLASS ID: pas utilisé

PARENT CLASS: top

ATTRIBUTES:

1	(m) Attribut-clé:	Implicit
2	(m) Attribut:	List of Ports
2.1	(m) Attribut:	Port
2.1.1	(m) Attribut:	MAU Type
2.1.2	(m) Attribut:	Autonegotiation State
2.1.3	(m) Attribut:	Link State

SERVICES:

1	(m) OpsService:	M_UNITDATA
2	(m) OpsService:	MAU Type Change
3	(m) OpsService:	Set MAU Type

6.3.11.2.3 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet IEEE 802.3 est adressé de façon implicite par le service.

List of Ports

Cet attribut doit être constitué des éléments suivants.

Port

Cet attribut est constitué des éléments suivants:

Type d'attribut: Unsigned8

MAU Type

Cet attribut contient la valeur du type courant de MAU selon l'IETF RFC 4836. Si la valeur de l'attribut Status est DOWN ou OFF, le type de MAU doit être zéro.

Type d'attribut: Unsigned16

Autonegotiation State

Cet attribut contient la valeur du réglage d'autonégociation courant. Pour l'émission optique, la valeur doit être mise à DISABLE.

Type d'attribut: Unsigned8

Valeurs permises: ENABLE, DISABLE

Link State

Cet attribut contient la valeur du réglage courant de statut de port. Il doit être OFF si l'unité de raccordement au support est désactivée. Il doit être DOWN si l'unité de raccordement au support est activée, mais aucune liaison (Link) n'est détectée. Il doit être UP si l'unité de raccordement au support est activée et une liaison (Link) est détectée.

Type d'attribut: Unsigned8

Valeurs permises: UP, DOWN, OFF

6.3.11.3 Spécification des services de l'IEEE 802.3

6.3.11.3.1 M_UNITDATA

Pour obtenir la définition de ce service, consulter l'IEEE 802.3.

Ce service local doit être utilisé pour émettre (M_UNITDATA.req) ou recevoir (M_UNITDATA.ind) une PDU.

6.3.11.3.2 MAU Type Change

Ce service local doit être utilisé pour indiquer un changement de certaines Local Systems Data (données de systèmes locaux). Le Tableau 95 montre les paramètres du service.

Tableau 95 – MAU Type change

Nom de paramètre	ind
Argument	M
Local Port ID	M
Local MAU Type	U
Local Autonegotiation State	U
Local Link State	U

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

Local Port ID

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local MAU Type

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Autonegotiation State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Link State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

6.3.11.3.3 Set MAU Type

Ce service local doit être utilisé pour indiquer un changement de certaines Remote Systems Data (données de systèmes distants). Le Tableau 96 montre les paramètres du service.

Tableau 96 – Set MAU Type

Nom de paramètre	ind
Argument	M
Local Port ID	M
Local MAU Type	U
Local Autonegotiation State	U
Local Link State	U

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

Local Port ID

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local MAU Type

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Autonegotiation State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

Local Link State

Ce paramètre est la valeur de l'attribut correspondant de l'objet ASE.

6.3.11.4 Invocation de l'objet IEEE 802.3

Pour l'invocation de l'objet IEEE 802.3, les règles suivantes s'appliquent:

- Des objets IEEE 802.3 doivent exister pour chaque interface.

6.3.12 ASE Fragmentation**6.3.12.1 Vue d'ensemble**

La longueur d'une période d'émission est limitée à l'extrémité inférieure de la taille maximale d'une trame Ethernet. La fragmentation réduit, avec une approche peer-to-peer, la taille acheminée d'une trame Ethernet et permet d'utiliser une LengthOfPeriod pouvant descendre à 31,25 µs.

L'ASE de fragmentation cache la fragmentation d'émission des couches supérieures en agissant comme une entité de service d'application intermédiaire et en assurant la même interface de service que l'ASE IEEE 802.3.

Pour cela, l'ASE Fragmentation est dérivé de l'ASE IEEE 802.3 et écrase le service opérationnel M_UNITDATA pour masquer les détails spécifiques à la fragmentation des couches supérieures.

Les trames devant être émises qui font l'objet d'une fragmentation seront fragmentées par le service opérationnel et émises en fragments en utilisant le service de base M_UNITDATA de l'ASE IEEE 802.3. Les trames fragmentées sont reçues par l'ASE IEEE 802.3, rassemblées pour restaurer la trame originale, puis indiquées par l'indication M_UNITDATA aux couches supérieures.

Les trames relatives aux protocoles PTCP, MRP et RTC (RT_CLASS_1, RT_CLASS_2 et RT_CLASS_3) ne sont jamais fragmentées.

L'ASE Fragmentation est facultatif; il doit être pris en charge, si le temps résiduel pour les trames NRT tombe en dessous de 125 µs.

6.3.12.2 Spécification de la classe Fragmentation**6.3.12.2.1 Généralités**

L'ASE Fragmentation définit un type d'objet Fragmentation.

6.3.12.2.2 Modèle

Un objet Fragmentation est décrit par le modèle suivant:

ASE: ASE Fragmentation

CLASS: Fragmentation

CLASS ID: pas utilisé

PARENT CLASS: ASE IEEE 802.3

ATTRIBUTS:

1 (m) Attribut-clé: Implicit

SERVICES:

1 (m) OpsService: M_UNITDATA

6.3.12.2.3 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet Fragmentation est adressé de façon implicite par le service.

6.3.12.3 Spécification du service Fragmentation

6.3.12.3.1 M_UNITDATA

Pour obtenir la définition de ce service, consulter l'IEEE 802.3.

Ce service local doit être utilisé pour émettre (M_UNITDATA.req) ou recevoir (M_UNITDATA.ind) une PDU.

6.3.12.4 Invocation de l'objet Fragmentation

Pour l'invocation de l'objet Fragmentation, les règles suivantes s'appliquent:

- Des objets Fragmentation doivent exister pour chaque interface.

6.3.13 ASE "IP suite"

6.3.13.1 Vue d'ensemble

L'utilisation des normes IETF RFC 768 (UDP), IETF RFC 791 (IP), IETF RFC 792 (ICMP), IETF RFC 826 (ARP), IETF RFC 1112 (IP Multicasting) est définie pour la présente spécification. Elle inclut des définitions pour les ports UDP, les adresses de multidiffusion IP et l'utilisation de champs d'en-tête IP pour RT sur UDP.

Cet ASE doit contenir des valeurs courantes volatiles de fonctionnement. Les valeurs permanentes en mémoire non volatile doivent être un sujet pour l'ASE DCP définie en 6.3.1.

6.3.13.2 Spécification de la classe "IP suite"

6.3.13.2.1 Généralités

L'ASE "IP suite" définit un type d'objet "Physical Device Management":

6.3.13.2.2 Modèle

Un objet IP suite est décrit par le modèle suivant:

ASE: ASE IP suite

CLASS: IP suite

CLASS ID: pas utilisé

PARENT CLASS: top

ATTRIBUTS:

1 (m) Attribut-clé: Implicit

2 (m) Attribut: IP

2.1 (m) Attribut: IP Address

2.2	(m) Attribut:	Subnet Mask
2.3	(m) Attribut:	Standard Gateway
3	(m) Attribut:	ARP
3.1	(m) Attribut:	Cache Size
3.2	(m) Attribut:	Cache Timeout
3.3	(m) Attribut:	List of Cache Entries
3.3.1	(m) Attribut:	IEEE 802.3 MAC Address
3.3.2	(m) Attribut:	IP Address
4	(m) Attribut:	List of IP Multicast addresses
4.1	(m) Attribut:	IP Multicast address

SERVICES:

1	(o) OpsService:	Set ARP Cache

6.3.13.2.3 Attributs**Implicit**

L'attribut clé « Implicit » indique que l'objet "IP suite" est adressé de façon implicite par le service.

IP

Cet attribut contient les attributs suivants:

IP Address

Cet attribut contient l'adresse IP actuelle conforme à l'IETF RFC 791.

Type d'attribut: Unsigned32

Subnet Mask

Cet attribut contient le masque de sous-réseau actuel conforme à l'IETF RFC 791.

Type d'attribut: Unsigned32

Standard Gateway

Cet attribut contient l'adresse IP actuelle de la passerelle normalisée selon l'IETF RFC 791.

Type d'attribut: Unsigned32

ARP

Cet attribut contient les attributs suivants:

Cache Size

Cet attribut contient la taille du cache ARP et doit contenir une valeur pour placer en cache tous les appareils auxquels l'appareil est connecté. La CEI 61784-2 définit une valeur recommandée.

NOTE 1 La taille du cache influence les performances du fonctionnement E/S.

Type d'attribut: Unsigned32

Cache Timeout

Cet attribut contient la valeur de temporisation pour le rafraîchissement de l'entrée de cache ARP (voir IETF RFC 826). Cet attribut influence le temps pour reconnaître que des couples adresse IP/adresse MAC ont été modifiés en cas de remplacement d'appareil. La CEI 61784-2 définit une valeur recommandée.

NOTE 2 Des entrées statiques au cache ARP pour des appareils E/S constituent une solution si elles sont prises en charge par la pile UDP/IP.

Type d'attribut: Unsigned32

List of Cache Entries

Cette liste d'attributs contient les attributs suivants:

IEEE 802.3 MAC Address

Cet attribut contient la valeur de l'adresse MAC IEEE 802 correspondant à l'adresse IP.

Type d'attribut: OctetString[6]

IP Address

Cet attribut contient l'adresse IP conforme à l'IETF RFC 791.

Type d'attribut: Unsigned32

List of IP Multicast addresses

Cette liste d'attributs contient les attributs suivants:

IP Multicast address

Cet attribut contient l'adresse de multidiffusion IP pour la relation de communication en multidiffusion. Il doit être défini selon l'IETF RFC 2365 et doit suivre la règle de construction liée au Frame ID indiqué dans le Tableau 97.

Type d'attribut: Unsigned32

Les valeurs permises avec l'adresse MAC et le Frame ID associés sont énumérées dans le Tableau 97.

Tableau 97 – IP Multicast address

IP multicast address	Adresse MAC de multidiffusion associée	Frame ID associé	Usage
239.192.248.0	01-00-5E-40-F8-00	0xF800	Utilisée pour les relations de communication en multidiffusion conjointement à RT_CLASS_UDP
239.192.248.1	01-00-5E-40-F8-01	0xF801	Utilisée pour les relations de communication en multidiffusion conjointement à RT_CLASS_UDP
à	à	à	Utilisée pour les relations de communication en multidiffusion conjointement à RT_CLASS_UDP
239.192.251.254	01-00-5E-40-FB-FE	0xFBFE	Utilisée pour les relations de communication en multidiffusion conjointement à RT_CLASS_UDP
239.192.251.255	01-00-5E-40-FB-FF	0xFBFF	Utilisée pour les relations de communication en multidiffusion conjointement à RT_CLASS_UDP

6.3.13.3 Spécification du service IP suite

6.3.13.3.1 Set ARP Cache

Ce service local peut être utilisé pour manipuler le cache ARP local chargeant des entrées. Le Tableau 98 montre les paramètres du service.

Tableau 98 – Set ARP Cache

Nom de paramètre	req	cnf
Argument	M	
Number of Entries	M	
List of Cache Entries	U	
IEEE 802.3 MAC Address	M	
IP Address	M	
Result(+)		S(=)
Result (-)		S(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

Number of Entries

Cette liste facultative de paramètres doit contenir le nombre d'entrées de cache (Cache Entries) qui suit. Le nombre ne doit pas dépasser la valeur de l'attribut Cache Size.

List of Cache Entries

Cette liste facultative de paramètres doit être constituée des paramètres suivants:

IEEE 802.3 MAC Address

Ce paramètre contient la valeur pour l'attribut d'ASE équivalent.

IP Address

Ce paramètre contient la valeur pour l'attribut d'ASE équivalent.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

6.3.14 ASE "Domain name system"**6.3.14.1 Vue d'ensemble**

L'utilisation de l'IETF RFC 1034 (DNS) est définie pour la présente spécification. Elle inclut l'utilisation et la syntaxe de contenu du nom de domaine.

6.3.14.2 Spécification de la classe DNS**6.3.14.2.1 Généralités**

L'ASE "DNS" définit un type d'objet DNS.

6.3.14.2.2 Modèle

Un objet DNS est décrit par le modèle suivant.

ASE:	ASE DNS
CLASS:	DNS
CLASS ID:	pas utilisé
PARENT CLASS:	top
ATTRIBUTES:	
1	(m) Attribut-clé: Implicit
2	(m) Attribut: DNS
2.1	(m) Attribut: Preferred DNS Server
2.2	(m) Attribut: Alternate DNS Server 1
2.3	(m) Attribut: Alternate DNS Server 2
2.4	(m) Attribut: Alternate DNS Server 3
SERVICES:	
1	(o) OpsService: DNS Get Host By Name

6.3.14.2.3 Attributs**Implicit**

L'attribut clé « Implicit » indique que l'objet DNS est adressé de façon implicite par le service.

DNS

Cet attribut contient les attributs suivants:

Preferred DNS Server

Cet attribut contient l'adresse IP du serveur DNS préféré selon l'IETF RFC 791.

Type d'attribut: Unsigned32

Alternate DNS Server 1

Cet attribut contient l'adresse IP du premier serveur DNS de substitution selon l'IETF RFC 791.

Type d'attribut: Unsigned32

Alternate DNS Server 2

Cet attribut contient l'adresse IP du second serveur DNS de substitution selon l'IETF RFC 791.

Type d'attribut: Unsigned32

Alternate DNS Server 3

Cet attribut contient l'adresse IP du troisième serveur DNS de substitution selon l'IETF RFC 791.

Type d'attribut: Unsigned32

6.3.14.3 Spécification de service DNS

Les services de DNS doivent être dérivés de l'IETF 1034 (DNS).

6.3.15 ASE "Dynamic host configuration"

6.3.15.1 Vue d'ensemble

L'utilisation de l'IETF RFC 2131 (DHCP) est définie pour la présente spécification. Elle inclut l'utilisation et la syntaxe de contenu de Client ID.

6.3.15.2 Spécification de la classe DHCP

6.3.15.2.1 Généralités

L'ASE "DHCP " définit un type d'objet DHCP:

6.3.15.2.2 Modèle

Un objet DHCP est décrit par le modèle suivant.

ASE:	ASE DHCP
CLASS:	DHCP
CLASS ID:	pas utilisé
PARENT CLASS:	top
ATTRIBUTES:	
1	(m) Attribut-clé: Implicit
SERVICES:	
1	(o) OpsService: DHCP Get IP

6.3.15.2.3 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet DHCP est adressé de façon implicite par le service.

6.3.15.3 Spécification des services de DHCP

Les services de DHCP doivent être dérivés de l'IETF RFC 2131 (DHCP).

6.3.16 ASE "Simple network management"

6.3.16.1 Vue d'ensemble

L'utilisation des normes IETF RFC 2674 (Ponts avec classes de trafic), IETF RFC 2737 (MIB 2), IETF RFC 2863 (IF MIB), IETF RFC 3418 (SNMP), IETF RFC 3621 (Puissance sur Ethernet MIB), IETF RFC 4836 (MAU MIB) est définie pour la présente spécification. Elle inclut l'utilisation de différentes bases d'informations de gestion (MIB). En outre, une amélioration du MIB LLDP est spécifiée dans la CEI 61158-6-10.

6.3.16.2 Spécification de la classe SNMP

6.3.16.2.1 Généralités

L'ASE "SNMP" définit un type d'objet SNMP:

6.3.16.2.2 Modèle

Un objet SNMP est décrit par le modèle suivant:

ASE: ASE SNMP

CLASS: SNMP

CLASS ID: pas utilisé

PARENT CLASS: top

ATTRIBUTES:

- 1 (m) Attribut-clé: Implicit
- 2 (m) Attribut: Type10 MIB
- 2.1 (m) Attribut: Enterprise number
- 2.2 (o) Attribut: List of Vendors
- 2.2.2 (m) Attribut: Vendor OUI
- 3 (m) Attribut: List of Interfaces
- 3.1 (m) Attribut: List of Ports
- 3.1.1 (m) Attribut: ifInOctets
- 3.1.2 (m) Attribut: ifOutOctets
- 3.1.3 (m) Attribut: ifInDiscards
- 3.1.4 (m) Attribut: ifOutDiscards
- 3.1.5 (m) Attribut: ifInErrors
- 3.1.6 (m) Attribut: ifOutErrors

SERVICES:

- 1 (o) OpsService: SNMP Get

6.3.16.2.3 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet SNMP est adressé de façon implicite par le service.

Type10 MIB

Cet attribut se compose des éléments suivants:

Enterprise number

Cet attribut contient le numéro d'entreprise comme identificateur de MIB. La valeur doit être conforme au Tableau 99.

Tableau 99 – Enterprise number

Valeur	Signification
24 686	Numéro d'entreprise pour MIB de type 10

List of Vendors

Cet attribut liste d'attributs est constitué des attributs suivants:

Vendor OUI

Cet attribut définit quelle extension complémentaire de MIB spécifique à un fournisseur existe.

Type d'attribut: Unsigned32

Les valeurs admissibles sont montrées dans le Tableau 100.

Tableau 100 – Vendor OUI

Valeur	Signification
Type10 OUI	Le Type10 définit les données structurées subséquentes.
Vendor OUI	Le vendeur définit les données structurées subséquentes.
Others (Autres)	Réservé

List of Interfaces

Cet attribut liste d'attributs est constitué des attributs suivants:

List of Ports

Cet attribut liste d'attributs est constitué des attributs suivants:

NOTE Noter que la valeur « 0 » pour les compteurs particuliers de statistiques est floue. Si l'attribut a la valeur « 0 », le compteur de statistiques peut ne pas être pris en charge, ou bien rien n'a été compté jusqu'à présent. Si, toutefois, les compteurs ifInOctets et ifOutOctets sont renvoyés avec la valeur « 0 », on peut en déduire que les autres compteurs, si à «0», ne sont également pas pris en charge.

ifInOctets

Cet attribut contient le nombre d'octets dans les trames MAC valides reçues sur cette interface, y compris l'en-tête MAC et FCS. Pour plus de détails voir l'IETF RFC 2020.

Type d'attribut: Unsigned32

Valeurs permises: 0 (qui signifie que le compteur de statistique n'est pas pris en charge), 0 à 0xFFFFFFFF (qui signifie le nombre d'octets)

ifOutOctets

Cet attribut contient le nombre d'octets émis dans les trames MAC valides sur cette interface, y compris l'en-tête MAC et FCS. Pour plus de détails voir l'IETF RFC 1573.

Type d'attribut: Unsigned32

Valeurs permises: 0 (qui signifie que le compteur de statistique n'est pas pris en charge), 0 à 0xFFFFFFFF (qui signifie le nombre d'octets)

ifInDiscards

Cet attribut contient le nombre de paquets Rx choisis pour être supprimés pour cause de conflits de ressources dans les instances supérieures. Pour plus de détails voir l'IETF RFC 1573.

Type d'attribut: Unsigned32

Valeurs permises: 0 (qui signifie que le compteur statistique n'est pas pris en charge), 0 à 0xFFFFFFFF (qui signifie le nombre de paquets Rx)

ifOutDiscards

Cet attribut contient le nombre de paquets Tx choisis pour être supprimés pour cause de conflits de ressources dans des instances comme un interrupteur intégré. Pour plus de détails voir l'IETF RFC 1573.

Type d'attribut: Unsigned32

Valeurs permises: 0 (qui signifie que le compteur statistique n'est pas pris en charge), 0 à 0xFFFFFFFF (qui signifie le nombre de paquets Tx)

ifInErrors

Cet attribut contient la somme des mauvaises trames reçues pour cette interface. Pour plus de détails voir l'IETF RFC 2020.

Type d'attribut: Unsigned32

Valeurs permises: 0 (qui signifie que le compteur de statistique n'est pas pris en charge), 0 à 0xFFFFFFFF (qui signifie la somme des mauvaises trames reçues)

ifOutErrors

Cet attribut contient les erreurs émises sur cette interface. Pour plus de détails voir l'IETF RFC 1573.

Type d'attribut: Unsigned32

Valeurs permises: 0 (qui signifie que le compteur de statistique n'est pas pris en charge), 0 à 0xFFFFFFFF (qui signifie les erreurs émises)

6.3.16.3 Spécification de service SNMP

Les services de SNMP doivent être dérivés de l'IETF RFC 3418 (SNMP).

6.3.17 ASE "DL mapping" commun

6.3.17.1 Vue d'ensemble

Cet ASE fournit une interface commune pour le mapping de DL.

6.3.17.2 Spécification de la classe DL Mapping

6.3.17.2.1 Généralités

L'ASE DL Mapping définit un type d'objet DL Mapping:

6.3.17.2.2 Modèle

Un objet DL Mapping est décrit par le modèle suivant:

ASE: ASE DL Mapping
CLASS: DL Mapping
CLASS ID: pas utilisé
PARENT CLASS: top
ATTRIBUTES:
 1 (m) Attribut-clé: Implicit
SERVICES:
 1 (m) OpsService: P Data
 2 (m) OpsService: N Data
 3 (m) OpsService: A Data
 4 (m) OpsService: C Data

6.3.17.2.3 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet DL Mapping est adressé de façon implicite par le service.

6.3.17.3 Spécification des services de DL Mapping

6.3.17.3.1 P Data

Ce service est utilisé pour transférer les données horodatées de l'appareil E/S vers l'appareil de commande E/S et vice versa. Le Tableau 101 montre les paramètres du service.

Tableau 101 – P Data

Nom de paramètre	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
T Stamp		M	
DA	M	M(=)	
SA	M	M(=)	

Nom de paramètre	req	ind	cnf
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
N SDU	M	M(=)	
Result(+)			S
AREP			M
Result (-)			S
AREP			M
Status			M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

D Port

Ce paramètre contient le Port ID souhaité.

T Stamp

Ce paramètre contient le marqueur temporel.

DA

Ce paramètre contient l'adresse de destination conforme à l'IEEE 802.3.

SA

Ce paramètre contient l'adresse source conforme à l'IEEE 802.3.

VLAN Prio

Ce paramètre contient la priorité conforme à l'IEEE 802.1Q.

VLAN ID

Ce paramètre contient le VLAN ID selon l'IEEE 802.1Q.

LT

Ce paramètre contient la priorité conforme à l'IEEE 802.3.

N SDU

Ce paramètre contient l'APDU.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Status

Ce paramètre contient la cause de l'erreur.

6.3.17.3.2 N Data

Ce service est utilisé pour transférer des données autres que RTA et RTC de l'appareil E/S vers l'appareil de commande E/S et vice versa. Le Tableau 102 montre les paramètres du service.

Tableau 102 – N Data

Nom de paramètre	req	ind	cnf

Nom de paramètre	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
N SDU	M	M(=)	
Result(+)			S
AREP			M
Result (-)			S
AREP			M
Status			M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

D Port

Ce paramètre contient le Port ID souhaité.

DA

Ce paramètre contient l'adresse de destination conforme à l'IEEE 802.3.

SA

Ce paramètre contient l'adresse source conforme à l'IEEE 802.3.

VLAN Prio

Ce paramètre contient la priorité conforme à l'IEEE 802.1Q.

VLAN ID

Ce paramètre contient le VLAN ID selon l'IEEE 802.1Q.

LT

Ce paramètre contient la priorité conforme à l'IEEE 802.3.

N SDU

Ce paramètre contient l'APDU.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Status

Ce paramètre contient la cause de l'erreur.

6.3.17.3.3 A Data

Ce service est utilisé pour transférer des données RTA de l'appareil E/S vers l'appareil de commande E/S et vice versa. Le Tableau 103 montre les paramètres du service.

Tableau 103 – A Data

Nom de paramètre	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
A SDU	M	M(=)	
Result(+)			S
AREP			M
Result (-)			S
AREP			M
Status			M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

D Port

Ce paramètre contient le Port ID souhaité.

DA

Ce paramètre contient l'adresse de destination conforme à l'IEEE 802.3.

SA

Ce paramètre contient l'adresse source conforme à l'IEEE 802.3.

VLAN Prio

Ce paramètre contient la priorité conforme à l'IEEE 802.1Q.

VLAN ID

Ce paramètre contient le VLAN ID selon l'IEEE 802.1Q.

LT

Ce paramètre contient la priorité conforme à l'IEEE 802.3.

A SDU

Ce paramètre contient l'APDU.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Status

Ce paramètre contient la cause de l'erreur.

6.3.17.3.4 C Data

Ce service est utilisé pour transférer des données RTC de l'appareil E/S vers l'appareil de commande E/S et vice versa. Le Tableau 104 montre les paramètres du service.

Tableau 104 – C Data

Nom de paramètre	req	ind	cnf
Argument	M	M(=)	
CREP	M	M(=)	
D Port	M	M(=)	
DA	M	M(=)	
SA	M	M(=)	
VLAN Prio	M	M(=)	
VLAN ID	M	M(=)	
LT	M	M(=)	
N SDU	M	M(=)	
APDU Status	M	M(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

D Port

Ce paramètre contient le Port ID souhaité.

DA

Ce paramètre contient l'adresse de destination conforme à l'IEEE 802.3.

SA

Ce paramètre contient l'adresse source conforme à l'IEEE 802.3.

VLAN Prio

Ce paramètre contient la priorité conforme à l'IEEE 802.1Q.

VLAN ID

Ce paramètre contient le champ VLAN conforme à l'IEEE 802.1Q.

LT

Ce paramètre contient la priorité conforme à l'IEEE 802.3.

C SDU

Ce paramètre contient l'APDU.

APDU Status

Ce paramètre contient le champ APDU Status.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Status

Ce paramètre contient la cause de l'erreur.

7 Modèle de communication pour périphérie décentralisée

7.1 Concepts

7.1.1 Exigences de l'utilisateur

Le système d'automation type est constitué d'un ou plusieurs d'automates programmables connectés avec des systèmes E/S au diagramme/au processus. A l'heure actuelle, un système E/S est surtout mis en œuvre comme un système décentralisé hiérarchique ou un ensemble de connexions simple point avec une interface électrique bien définie telle que signaux numériques 24 V ou signaux analogiques 4 mA à 20 mA.

Ce modèle de communication définit un système décentralisé et la connexion repose sur la technologie de l'information et de la télécommunication de l'IEEE 802.3. Le modèle d'objet défini est une amélioration du modèle d'objet de la CEI 61158-5-3.

7.1.2 Caractéristiques intrinsèques

En combinant la structure définie dans la CEI 61131-1, les avantages suivants sont réalisés:

- Des changements dans la centrale n'induisent que de faibles changements de câblage.
- Des signaux critiques peuvent être transférés sur de courtes distances jusqu'à l'appareil de terrain distant (capteur/actionneur) et sur de longues distances à destination/en provenance de l'automate programmable avec une exactitude élevée.
- La paramétrisation et le diagnostic peuvent être réalisés sans câblage supplémentaire.
- Mise en service de sous-parties sans installations supplémentaires.

Le Tableau 105 donne une vue d'ensemble des exigences et des caractéristiques du système.

Tableau 105 – Exigences et caractéristiques

Exigences	Caractéristique intrinsèque
Court temps de réaction	Echange cyclique de plus de 1 000 entrées et sorties avec 32 appareils de terrain en moins de 1 ms
Le système d'automation est constitué d'un ou plusieurs automates programmables	Fonctionnement avec un seul appareil de commande ou plusieurs appareils de commande, il est possible d'associer à un seul appareil E/S plusieurs appareils de commande E/S
Simple appareil de terrain	Protocole simple, interface de communications à faible coût
Adaptation de modèle d'application de la CEI 61158-5-3/ CEI 61158-6-3	Modèle similaire d'objet de processus d'application avec mode de compatibilité
Protocoles unifiés	Utilisation du même mécanisme de protocole de transport temps réel pour différents modèles d'applications
Excellent diagnostic	appareils
Appareils de terrain intelligents	Communication acyclique qui fournit un plan flexible et évolué d'adressage des données au sein des appareils E/S. La possibilité de transférer une alarme de l'appareil E/S vers l'appareil de commande E/S et vice vers avec un acquittement explicite
Communication efficace entre appareils E/S et entre appareil de commande E/S et appareils E/S	Mécanisme Fournisseur/Consommateur
Synchronisation d'applications	Mode isochrone, gigue inférieure à 1 µs, signalisation à l'application
Synchronisation d'horloges	Synchronisation du temps à haute précision
Redondance	Redondance d'appareil E/S et d'appareil de commande E/S, prise en charge de la redondance de supports

Exigences	Caractéristique intrinsèque
Interopérabilité	Définitions précises et complète, y compris la définition du comportement du système
Modification au cours du fonctionnement	Reconfiguration dynamique

7.1.3 Associations

Le modèle de communication prend en charge trois types d'associations qui sont caractérisées par des types d'appareils impliqués dans l'association.

NOTE 1 Des associations complémentaires, par exemple téléchargement vers l'amont ou vers l'aval, peuvent être possibles, mais ne font pas partie du modèle de communication.

Le modèle de communication prend en charge la communication d'appareils de terrain avec un ou plusieurs appareils de commande (par exemple: automate programmable ou système de commande distribué) par l'intermédiaire d'associations (voir Figure 9). Il inclut l'échange de données cycliques relatif à des données E/S et à l'émission d'alarmes, l'émission de données de paramètres, de données de configuration, de données d'identification, de données de journaux, de données de diagnostic et de données d'enregistrement.

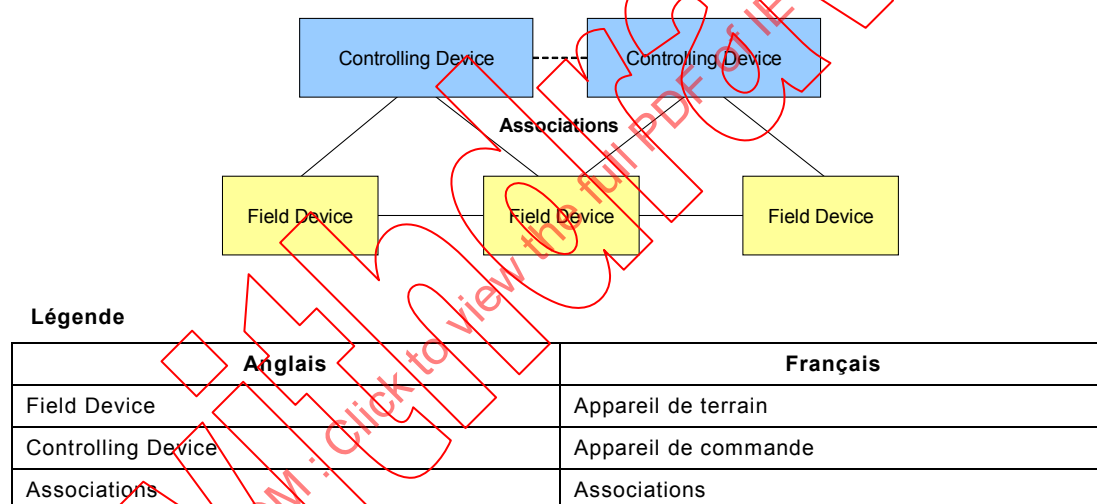
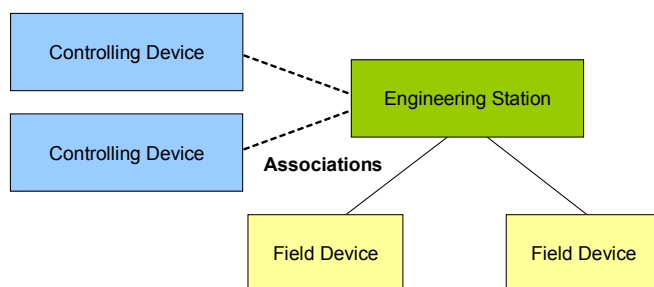


Figure 9 – Exemple de communication entre appareils de commande et appareils de terrain

En outre, la communication entre un appareil d'ingénierie et plusieurs appareils de commande ou de terrain est également prise en charge (voir Figure 10). L'association entre appareils d'ingénierie et appareils de terrain inclut l'émission de données de paramètres, de données de configuration, de données d'identification, de données de journaux (LogBook), de données de diagnostic et de données d'enregistrement au moyen du protocole RPC sans connexion.

NOTE 2 L'association entre appareil d'ingénierie et appareil de terrain inclut l'émission de diagnostic d'appareil de commande, des domaines et des commandes pour l'activation d'appareil au moyen du protocole RPC sans connexion (pour complément d'étude).

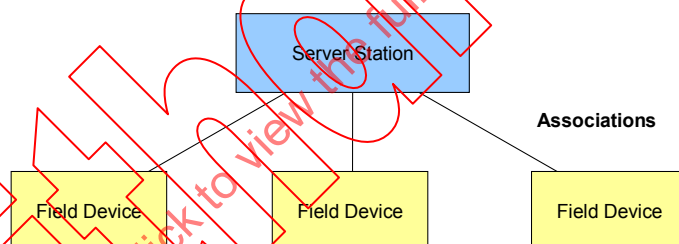


Légende

Anglais	Français
Field Device	Appareil de terrain
Controlling Device	Appareil de commande
Engineering Station	Station d'ingénierie
Associations	Associations

Figure 10 – Exemple de communication entre une station d'ingénierie et plusieurs appareils de commande et de terrain

En plus de ces modèles de communication, il prend également en charge la communication entre une station serveur et des appareils de terrain (voir Figure 11). L'association entre une station serveur et un appareil de terrain inclut l'émission de données d'enregistrement au moyen du protocole RPC sans connexion.

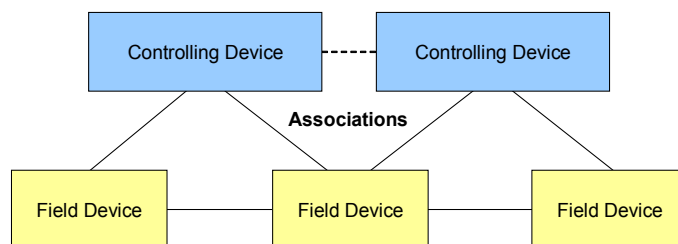


Légende

Anglais	Français
Field Device	Appareil de terrain
Server Station	Station serveur
Associations	Associations

Figure 11 – Exemple de communication entre appareils de terrain et une station serveur

En plus de ces modèles de communication, il prend également en charge la communication entre des appareils de terrain (voir Figure 12). Ce modèle de communication inclut un appareil de commande pour établir et commander l'association entre les appareils de terrain.

**Légende**

Anglais	Français
Field Device	Appareil de terrain
Server Station	Station serveur
Associations	Associations

Figure 12 – Exemple de communication entre appareils de terrain

NOTE 3 L'association entre appareils de commande peut inclure, par exemple, l'émission de données E/S, de diagnostic, d'alarmes et de paramètre.

7.1.4 Types d'appareils**7.1.4.1 Généralités**

Un appareil d'automation physique peut héberger un ou plusieurs des mêmes appareils ou appareils différents avec des associations appropriées.

7.1.4.2 Appareil de commande E/S

L'appareil de commande E/S est un appareil de commande qui est associé à un ou plusieurs appareils E/S (appareils de terrain). L'appareil de commande E/S accomplit l'une ou plusieurs des fonctionnalités suivantes:

- a) fonctionnalités cycliques
 - échange de données E/S avec des appareils E/S connexes
- b) fonctionnalités acycliques
 - lit le diagnostic à partir d'appareils E/S
 - configuration d'appareils E/S
 - écrit des données de paramètres dans les appareils E/S (paramètres de démarrage ou d'application)
 - traitement de demandes de configuration et de diagnostic d'un appareil d'ingénierie
 - établir le contexte aux appareils E/S au moyen de la gestion de contexte
 - accès acyclique à des données d'enregistrement d'appareils E/S
 - traitement d'alarmes issues d'appareils E/S
 - envoi d'alarmes à des appareils E/S
- c) fonctionnalités générales
 - redondance
 - reconfiguration dynamique
 - fonctionnement isochrone
 - communication de fournisseur/consommateur entre appareils E/S et appareils de commande E/S

- synchronisation de l'horloge

7.1.4.3 Superviseur E/S

Le superviseur E/S (IO supervisor) est un appareil d'ingénierie qui gère la fourniture de données de configuration (jeux de paramètres) et le recueil de données de diagnostic à destination/en provenance d'appareils de commande E/S et/ou d'appareils E/S.

7.1.4.4 Serveur de paramètre E/S

Le serveur de paramètres E/S est une station serveur qui sert à stocker et à charger des données de configuration d'application (objets de données d'enregistrement) issues d'appareils E/S (client).

7.1.4.5 Appareil E/S

7.1.4.5.1 Généralités

L'appareil E/S est un appareil de terrain et accomplit les activités suivantes, selon la fonctionnalité.

Les fonctionnalités sont:

- a) fonctionnalités cycliques
 - échange de données E/S avec un ou plusieurs appareils de commande E/S assignés
 - échange de données E/S avec des appareils E/S connexes
- b) fonctionnalités acycliques
 - dispositions pour la fourniture de données de diagnostic
 - traitement de demandes de configuration issues d'un appareil de commande E/S
 - fourniture d'un accès acyclique à des données d'enregistrement pour un appareil de commande E/S
 - fourniture d'alarmes à l'appareil de commande E/S assigné
 - traitement de paramètres (paramètre de démarrage ou d'application)
 - traitement de demandes de configuration et de diagnostic d'un appareil d'ingénierie
 - traitement d'alarmes issues d'appareils de commande E/S
 - envoi d'alarmes à des appareils de commande E/S
- c) fonctionnalités générales
 - communication de fournisseur/consommateur entre appareils E/S et appareils de commande E/S
 - redondance
 - reconfiguration dynamique
 - fonctionnement isochrone
 - synchronisation de l'horloge

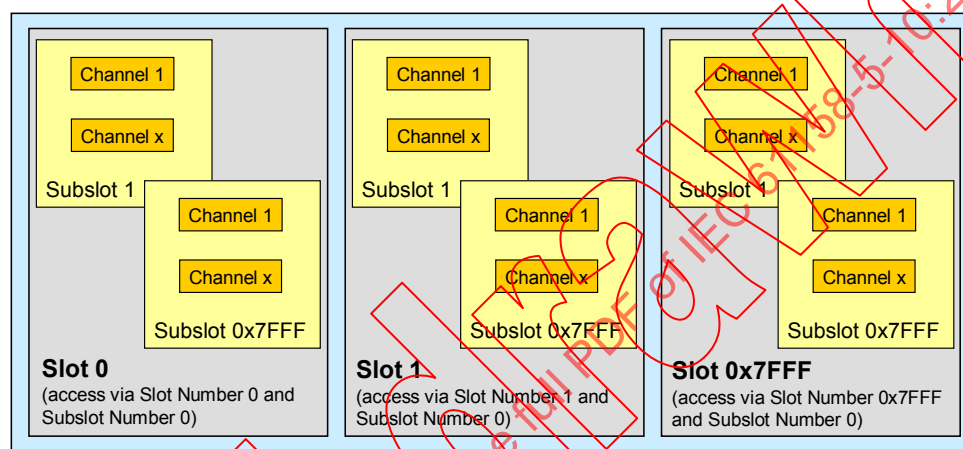
Des classes de conformité d'appareils E/S sont définies pour prendre en charge des fonctions communes.

En général, un appareil E/S est une composition hiérarchique comprenant les éléments suivants:

- une ou plusieurs instances d'appareils E/S (voir Annexe A);
- chaque instance d'appareil E/S inclut un ou plusieurs processus d'application référencés par leur identificateur (API);

- chaque API inclut une ou plusieurs baies (slot);
- chaque baie inclut une ou plusieurs sous-baies (subslot);
- chaque sous-baie peut inclure une ou plusieurs voies.

L'appareil E/S est composé de différentes unités de structure pour regrouper des objets de processus d'application et fournit un certain niveau d'abstraction. Ces unités de structure peuvent refléter des composants matériels ou des unités fonctionnelles virtuelles de l'appareil de terrain. Le but est de fournir un jeu adapté de paramètres d'adresse. Ces unités de structure sont appelées des «baies», «sous-baies» et «voies» qui peuvent également refléter des unités physiques, des sous-unités ou un seul point de connexion d'un appareil E/S. La Figure 13 montre la structure pour un API arbitraire. L'accès aux instances d'unités de structure, API, baie et sous-baie est effectué au moyen de l'Object UUID, de l'API Number, du Slot Number et du Subslot Number dans le modèle d'adresses.



Légende

Anglais	Français
(access via Slot Number 0 and Subslot Number 0)	(accès par le numéro de baie 0 et le numéro de sous-baie 0)
(access via Slot Number 0x7FFF and Subslot Number 0)	(accès par le numéro de baie 0x7FFF et le numéro de sous-baie 0)
(access via Slot Number 1 and Subslot Number 0)	(accès par le numéro de baie 1 et le numéro de sous-baie 0)
Subslot 0x7FFF	Sous-baie 0x7FFF
Channel 1	Voie 1
Channel x	Voie x
Slot 0	Baie 0
Slot 0x7FFF	Baie 0x7FFF
Slot 1	Baie 1
Subslot	Sous-baie
Subslot 1, e.g. "DAP"	Sous-baie 1, par exemple: "DAP"

Figure 13 – Unités structurales d'un API arbitraire d'un appareil E/S (général)

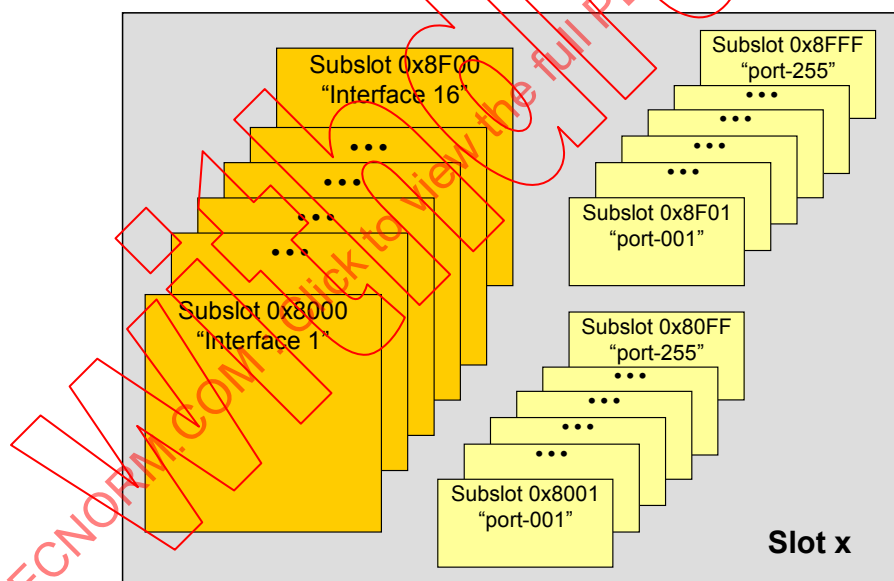
En outre, les sous-baies 0 à 0x7FFF peuvent être utilisées dans tous les API d'un appareil E/S en parallèle. Une relation entre applications particulière peut être établie à plus d'un processus d'application. Une instance d'appareil E/S avec plus d'un identificateur de processus d'application (API) est appelée «Multi-API-Instance» (c'est-à-dire instance à plusieurs API).

NOTE Une instance d'appareil E/S prenant en charge un API de profil particulier est une Multi-API-Instance, car la prise en charge de l'API 0 de l'instance d'appareil E/S est toujours obligatoire.

Le Subslot Number 0 doit être utilisé pour représenter le module dans le cas spécial «pull module», si l'alarme «module» est prise en charge. Dans ce cas, l'alarme «pull module» est simulée avec une alarme «pull submodule» sur la sous-baie 0. Il n'est alors pas possible d'avoir une Subslot 0 «real» contenant les voies E/S, diagnostic ou données d'enregistrement.

Si l'alarme «pull module» est prise en charge, un «réel» Subslot 0 existe, et Subslot 0 peut contenir des voies E/S, des données de diagnostic ou des données d'enregistrement.

En outre, des sous-baies spéciales dans la plage de 0x8000 à 0x8FFF sont définies pour l'API 0. Un appareil E/S doit contenir au moins une interface avec des ports. Par conséquent, au moins un Slot (par exemple Slot Number 0) doit contenir jusqu'à 16 autres Subslots spéciaux de 0x8000 (Interface 1), 0x8100 (Interface 2), à 0x8F00 (Interface 16) appelés "Interface Subslots" (sous-baies d'interface). Les Interface Subslot définissent l'accès distant à des paramètres d'adresse IP et le nom de l'interface (nom de DNS). En outre, chaque Interface Subslot doit avoir au moins un port assigné et au maximum 255 ports assignés. Ces sous-baies de 0x8i01 (Port 1) à 0x8iFF (Port 255) sont appelées "Port Subslot" (sous-baies de port). La sous-baie définit l'accès distant au paramètre spécifique à un port, par exemple Own Port ID. La Figure 14 décrit un exemple avec plusieurs ports sur la même baie appartenant à une seule interface. Toutes ces sous-baies particulières doivent seulement être utilisées dans l'API 0. Les valeurs "port-001" à "port-255" doivent être utilisées pour nommer les sous-modules de port si tous les sous-modules de port sont situés dans la même baie que le sous-module d'interface.

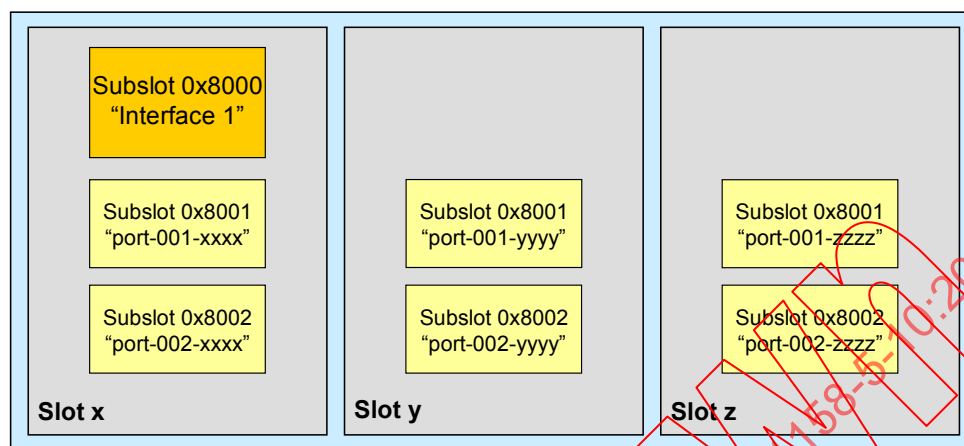


Légende

Anglais	Français
Subslot 0x8000, "Interface 1"	Sous-baie 0x8000, "Interface 1"
Subslot 0x8001, "port-001"	Sous-baie 0x8001, "port-001"
Slot x	Baie x
Subslot 0x8F00, "Interface 16"	Sous-baie 0x8F00, "Interface 16"
Subslot 0x8F01, "port-001"	Sous-baie 0x8F01, "port-001"
Subslot 0x80FF, "port-255"	Sous-baie 0x80FF, "port-255"
Subslot 0x8FFF, "port-255"	Sous-baie 0x8FFF, "port-255"

Figure 14 – Exemple 1 d'unités structurales pour interfaces et ports au sein de l'API 0

La Figure 15 représente un exemple dans lequel plusieurs sous-modules de port appartenant à une interface sont situés sur plusieurs baies. Les valeurs "port-001-xxxx" à "port-255-zzzz" doivent alors être utilisées pour nommer les sous-modules de port, tandis que "xxxx" désigne le numéro de baie avec des "0" de tête (par exemple port-002-00005").



Légende

Anglais	Français
Subslot 0x8000, "Interface 1"	Sous-baie 0x8000, "Interface 1"
Subslot 0x8001, "port-001-xxxx"	Sous-baie 0x8001, "port-001-xxxx"
Slot x	Baie x
Subslot 0x8002, "port-002-xxxx"	Sous-baie 0x8002, "port-002-xxxx"
Subslot 0x8001, "port-001-yyyy"	Sous-baie 0x8001, "port-001-yyyy"
Slot y	Baie y
Subslot 0x8002, "port-002-yyyy"	Sous-baie 0x8002, "port-002-yyyy"
Subslot 0x8001, "port-001-zzzz"	Sous-baie 0x8001, "port-001-zzzz"
Slot z	Baie z
Subslot 0x8002, "port-002-zzzz"	Sous-baie 0x8002, "port-002-zzzz"

Figure 15 – Exemple 2 d'unités structurales pour interfaces et ports au sein de l'API 0

Il convient que toutes les configurations possibles d'un appareil E/S soient décrites dans le fichier GSD de l'appareil.

Dans le cas spécial de redondance du système, le schéma de numérotation des sous-baies pour les interfaces et les ports est étendu.

Dans ce cas spécial seulement

- Le numéro de sous-baie 0x8000 à 0x8FFF désigne une sous-baie utilisée pour l'interface ou le port situé à gauche ou en-dessous du nœud de transmission.
- Le numéro de sous-baie 0x9000 à 0x9FFF désigne une sous-baie utilisée pour l'interface ou le port situé à droite ou au-dessus du nœud de transmission.

Pour la plage de numéros de sous-baie 0x9000 à 0x9FFF, les mêmes règles s'appliquent que pour la plage 0x8000 à 0x8FFF; par exemple 0x9i00 pour un sous-module d'interface avec « i » dans la plage de 0 à 15, désignant le numéro de l'interface; 0x9ipp pour un sous-module de port avec « pp » dans la plage de 1 à 255, désignant le numéro du port.

7.1.4.5.2 Slot

La couche application utilise des baies pour refléter la structuration des fonctions ou des composants (par exemple: module matériels, unités logiques) à l'intérieur d'un appareil E/S. Une baie doit avoir un ou plusieurs sous-baies qui représentent un niveau supplémentaire pour structurer les données.

Chaque baie doit être adressée par un numéro de baie commençant par zéro. Le dernier Slot Number possible est 0x7FFF. La numérotation peut comporter des trous.

NOTE Le modèle de baie est un modèle abstrait d'adresses utilisant des références numériques. Les modules matériels réels ou les unités fonctionnelles virtuelles peuvent même occuper plus d'une baie. Les caractéristiques de l'appareil relatives à l'utilisation de baies sont définies dans une description générale de station à des fins de configuration.

En outre, chaque module possède des données de configuration. Au cours de la phase d'établissement de connexion, la configuration demandée doit être comparée aux données de configuration réelle. Si une baie n'est pas occupée par un module physique ou virtuel, elle ne fait pas partie de la configuration.

Dans les appareils E/S modulaires, les baies déterminent la configuration réelle de l'appareil de terrain. Le nombre de baies est fixé en fonction de la spécification de l'appareil.

Un appareil E/S compact est un cas particulier d'appareil E/S modulaire avec une ou plusieurs configurations fixes.

7.1.4.5.3 Sous-baie

La couche application utilise des sous-baies pour refléter la structuration des fonctions ou d'autres composants (par exemple: unités matérielles, unités logiques) à l'intérieur d'une baie d'un appareil E/S. Une sous-baie peut avoir une ou plusieurs voies qui représentent la structure réelle de «Input and/or Output Data» (données d'entrée et/ou de sortie). Ces voies peuvent être une subdivision supplémentaire de l'objet «Input and/or Output Data».

Chaque sous-baie sera adressée par un numéro de sous-baies compris dans la plage 0 à 0x7FFF. En outre, dans une baie, des sous-baies particulières 0x8000 à 0x8FFF pour des interfaces et ports connexes peuvent exister; dans le cas spécial de redondance du système, les sous-baies 0x9000 à 0x9FFF pour l'interface et les ports associés situés à droite ou au-dessus du nœud final de transmission peuvent exister.

Si `ARProperties.PullModuleAlarmAllowed` n'est pas utilisé, la sous-baie numéro zéro est utilisée conjointement à `AlarmType` «Pull» pour adresser un certain module dans l'appareil E/S. Dans ce cas, elle ne doit pas contenir de voies et de données E/S.

Si `ARProperties.PullModuleAlarmAllowed` est utilisé, la sous-baie numéro zéro agit comme une sous-baie réelle. Par ailleurs, les sous-baies réelles peuvent contenir des voies et des données E/S. L'utilisation réelle d'un numéro de sous-baie et d'un indice pour adresser des données au sein de l'appareil est spécifique à un fabricant.

Subslot 0 adresse le module dans le domaine d'application de l'alarme «Pull Alarm» pour signaler le tirage d'un module, si `ARProperties.PullModuleAlarmAllowed` n'est pas utilisé. Si `ARProperties.PullModuleAlarmAllowed` est utilisé, Subslot 1 à 0x7FFF est librement utilisable dans le domaine d'application de l'AP adressé.

Au sein d'une baie, une sous-baie fournit un ou plusieurs des objets suivants:

- Record Data c'est-à-dire" données d'enregistrement "(données avec sémantique ouverte, par exemple un paramètre), adressé par
 - numéro de baie (0 à 0x7FFF),

- numéro de sous-baie (0 à 0x9FFF ou autrement 1 à 0x9FFF, en fonction de ARProperties.PullModuleAllowed), et
- indice (0 à 0xFFFF).
- IO Data (c'est-à-dire: Données E/S) comme une structure de données structurées E/S - attendu que la structure dépend de la description générale de station, chaque structure de données E/S inclut des informations de statuts- adressé par
 - numéro de baie (0 à 0x7FFF), et
 - numéro de sous-baie (0 à 0x7FFF ou autrement 1 à 0x7FFF, en fonction de ARProperties.PullModuleAllowed).

NOTE Les sous-baies particulières utilisées pour l'interface et les ports connexes ne contiennent pas de données E/S.

- Diagnosis (Diagnostic) contenant le diagnostic de voie ou le diagnostic générique, pour identification plus poussée, et adressé par
 - numéro de baie (0 à 0x7FFF),
 - numéro de sous-baie (0 à 0x9FFF ou autrement 1 à 0x9FFF, en fonction de ARProperties.PullModuleAllowed), et
 - indice (établi implicitement par le service de diagnostic),
- Alarms (alarmes), adressé par
 - numéro de baie (0 à 0x7FFF),
 - numéro de sous-baie (0 à 0x9FFF ou autrement 1 à 0x9FFF, en fonction de ARProperties.PullModuleAllowed), et
 - type d'alarme.

7.1.4.5.4 Channel (c'est-à-dire voie, canal)

La couche application utilise des baies et des sous-baies pour refléter la structuration des fonctions et des composants (par exemple: unités matérielles, unités logiques) dans un appareil E/S. Une baie/sous-baie peut avoir une ou plusieurs voies qui représentent la structure réelle de "Input and/or Output Data" (données d'entrée et/ou de sortie). Ces voies peuvent être une subdivision supplémentaire de l'objet Input and/or Output Data.

Les voies sont identifiées dans des informations de diagnostic ou d'alarme seulement.

7.1.4.5.5 Hiérarchie d'identification

La hiérarchie d'identification suivante est définie pour les éléments de la couche application:

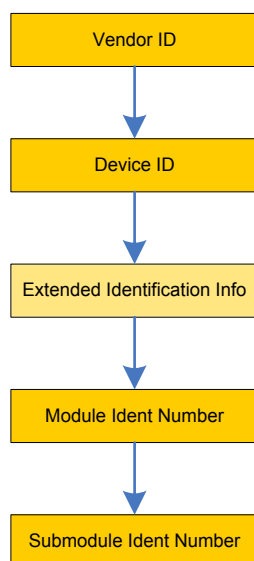


Figure 16 – Hiérarchie d'identification

Le Vendor ID est un numéro d'organisation globalement administré qui identifie un fabricant particulier; le Device ID est affecté de manière unique au type d'appareil par le fabricant de l'appareil. Voir également 7.6.1.

Le bloc Extended Identification Info permet d'étendre les règles d'identification pour le profil commun « Controller to Controller Communication ». Le bloc Extended Identification Info est fourni sous la forme d'un bloc AR Vendor Block pendant le service Connect de l'ASE Context. Si aucune identification étendue n'est fournie, aucune règle d'identification additionnelle ne s'applique.

Le Module Ident Number doit désigner de manière unique l'ensemble des modules qui peuvent apparaître dans un appareil spécifique. Si un module peut être placé dans différents appareils, le ModuleIdentNumber doit être affecté de manière unique parmi l'ensemble de ces appareils.

Le Submodule Ident Number doit désigner de manière unique l'ensemble des sous-modules qui peuvent apparaître dans un module spécifique. Au-delà de l'identification du matériel du sous-module, il identifie également la représentation des données d'entrée et/ou de sortie du module en ce qui concerne sa longueur et son type de données. Si un sous-module peut être placé dans différents modules, le SubmoduleIdentNumber doit être affecté de manière unique parmi l'ensemble de ces modules.

Si le même sous-module matériel possède différentes représentations des données d'entrée et/ou de sortie, le sous-module doit posséder différents Submodule Ident Numbers pour chacune de ces représentations. Noter que chacun de ces Submodule Ident Numbers doit toujours être capable d'identifier le sous-module spécifique, ce qui implique que chacun de ces numéros d'identification doive être différent des numéros d'identification des autres matériels de sous-module sur le même module.

Par conséquent, les paramètres Vendor ID, Device ID et Module Ident Number devant avoir une relation un-à-un (ce qui signifie qu'il existe exactement un numéro d'identification pour un type de matériel particulier), un sous-module peut posséder des SubmoduleIdentNumbers uniques.

Les numéros d'identification sont stockés dans le matériel et peuvent être lus, par exemple via l'UUID de l'objet RPC (Vendor ID et Device ID), via les fonctions I&M (Vendor ID et Device

ID), via les données d'AR (Extended Identification Info) et via les données d'identification réelles (Module Ident Number et Submodule Ident Number).

7.1.4.5.6 Objet combiné

Un appareil E/S peut définir des « objets combinés ». Un objet combiné particulier est un groupe de sous-modules associés, par exemple une interface et ses ports. L'un des sous-modules se définit comme un « point d'accès » de l'objet combiné, par exemple l'interface.

Chaque sous-module d'un objet combiné particulier doit appartenir à la relation d'application.

Un objet combiné sera paramétré en écrivant le conteneur CO sur son point d'accès; adressé par une API, une baie, une sous-baie et un indice. Le conteneur CO contient le paramétrage de l'ensemble des sous-modules; pour ce faire, il est divisé en sous-blocs; chacun possédant l'adresse du sous-module (API, baie et sous-baie), l'indice de l'enregistrement de données à écrire ainsi que les données proprement dit.

Le point d'accès achemine les objets de données d'enregistrement qui sont stockés dans le conteneur CO jusqu'aux sous-modules de l'objet combiné, pendant le paramétrage. Si un sous-module n'est pas disponible à ce moment, le point d'accès conserve les données et les achemine jusqu'au sous-module, dès qu'il redevient disponible. Si un sous-module particulier est connecté par exemple, il informe l'appareil de commande E/S comme d'habitude au moyen d'une alarme de connexion; néanmoins, comme l'appareil de commande E/S comporte uniquement les informations de paramétrage se rapportant au point d'accès de l'objet combiné (autrement dit le conteneur CO), le paramétrage de ce sous-module à partir de l'appareil de commande E/S est vide.

Cette procédure permet de connecter un lot de sous-modules en même temps (ex: une interface enfichable avec ses propres ports) et d'avoir un support de paramétrage implicite pour l'ensemble des sous-modules ensemble (ce qui est particulièrement important dans le contexte de l'appareil physique). Cela ne fonctionne néanmoins que si toutes les données d'enregistrement pour l'objet combiné sont placées dans le conteneur d'objet combiné.

Un objet combiné n'a pas d'impact sur l'appareil de commande E/S, de son point de vue, seul le point d'accès CO détient les paramètres; certains sous-modules possèdent des paramètres, d'autres non.

Les données d'enregistrement sont toujours lues directement à partir du sous-module particulier; le conteneur CO n'est pas lisible.

Un cas particulier pour l'objet combiné est utilisé pour l'appareil physique (ex: dans le contexte d'interfaces et de ports enfichables). Le point d'accès CO dans ce cas est le sous-module d'interface. Le conteneur CO contient le paramètre de l'ensemble des sous-modules d'interface et de port.

7.1.4.5.7 Données d'identification et de maintenance

Ces données identifient un sous-module et fournissent des informations complémentaires pour supporter l'utilisateur final dans différents scénarios d'un cycle de vie de sous-modules (configuration, mise en service, paramétrage, diagnostic, maintenance, réparation, mises à jour du firmware, gestion de biens, piste d'audit entre autres).

Pour ce faire, on utilise une unité de commande. Une unité de commande peut néanmoins se composer de plusieurs sous-modules (ex: un module constitué d'un sous-module d'interface, un lot de sous-modules de port, ainsi que d'autres sous-modules). Une telle unité de commande possède juste des données d'identification et de maintenance distinctes.

Exactement l'un des sous-modules de l'unité de commande est responsable du transport des données d'identification et de maintenance. Ce sous-module est appelé « porteuse ». Seule la porteuse permet l'accès en écriture aux données d'identification et de maintenance.

Pour permettre à chaque relation d'application d'accéder aux données d'identification et de maintenance d'un sous-module particulier (ex: appareil partagé), chaque sous-module doit prendre en charge la lecture des données d'identification et de maintenance. Il joue le rôle de proxy avec la porteuse et renvoie les données d'identification et de maintenance stockées sur la porteuse.

Les règles suivantes s'appliquent aux données de filtre I&M (voir 7.3.6.3.18):

- La liste des unités de commande est renvoyée avec les données de porteuse I&M0; cette liste répertorie les sous-modules jouant le rôle de porteuse pour les données I&M.
- Il existe au moins un sous-module, où I&M1, I&M2, I&M3 et I&M4 peuvent être écrits. Ce sous-module est indiqué dans la liste représentative des appareils.

7.1.5 Modèle d'instance et adresses d'appareils

La conception et la construction d'appareils physiques d'automatisation ne relèvent pas du domaine d'application de la présente spécification et sont donc spécifiques à un fournisseur. Cependant, l'Annexe A fournit un modèle pour appareils physiques pour héberger un ou plusieurs appareils de commande E/S ou appareils E/S.

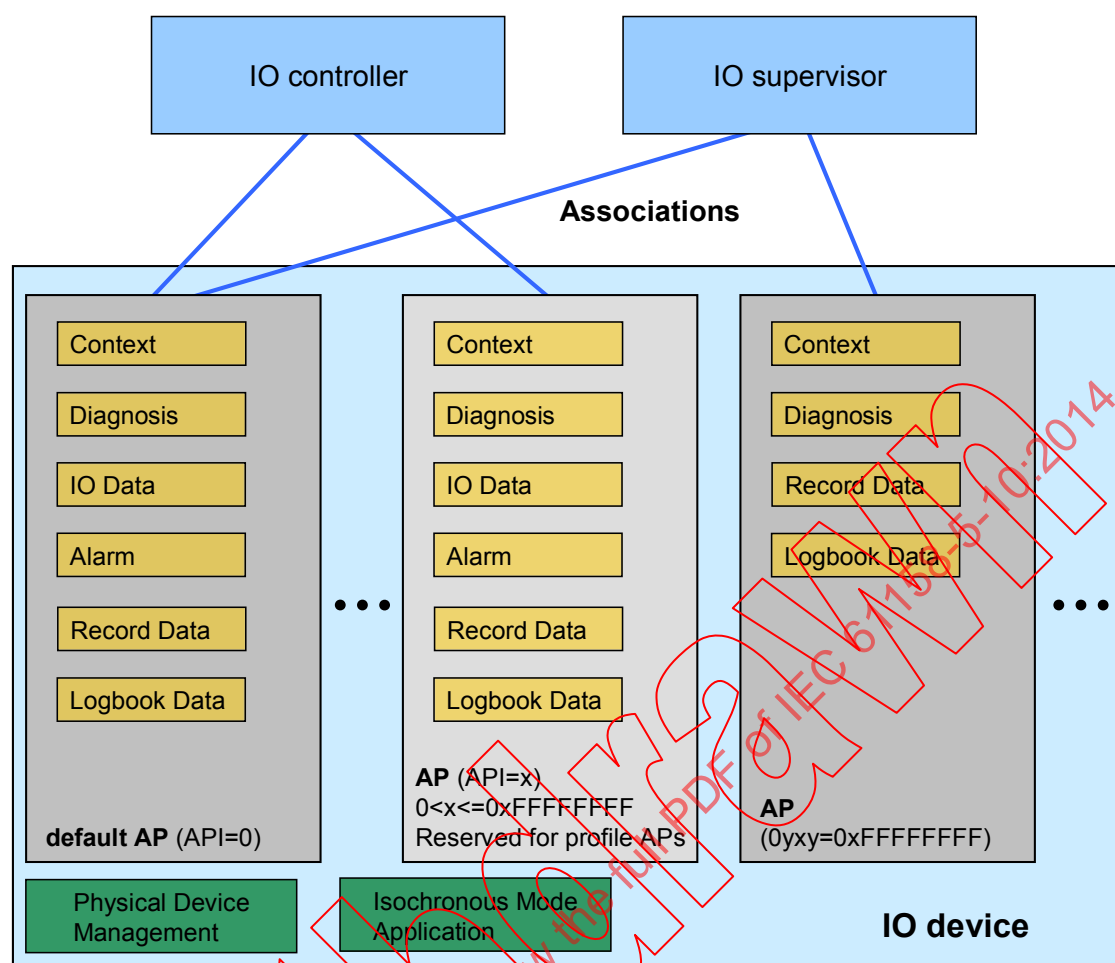
7.1.6 Processus d'application

7.1.6.1 Vue d'ensemble

Dans l'environnement de processus d'application, une application peut être partitionnée et distribuée en un certain nombre d'appareils sur le réseau. Chacune de ces partitions est appelée "processus d'application" (AP pour «application process»). Un appareil peut avoir plusieurs AP (voir Figure 17). Dans ce cas, chaque AP individuel est identifié de façon univoque par un identificateur d'AP (API pour «AP Identifier»).

Chaque appareil E/S doit prendre en charge un Default AP («AP par défaut») avec API = 0.

Les Default AP doivent donner des informations relatives à l'appareil. D'autres AP sont réservés pour être affectés de façon univoque à des profils d'appareil et à une utilisation normalisée ultérieure.

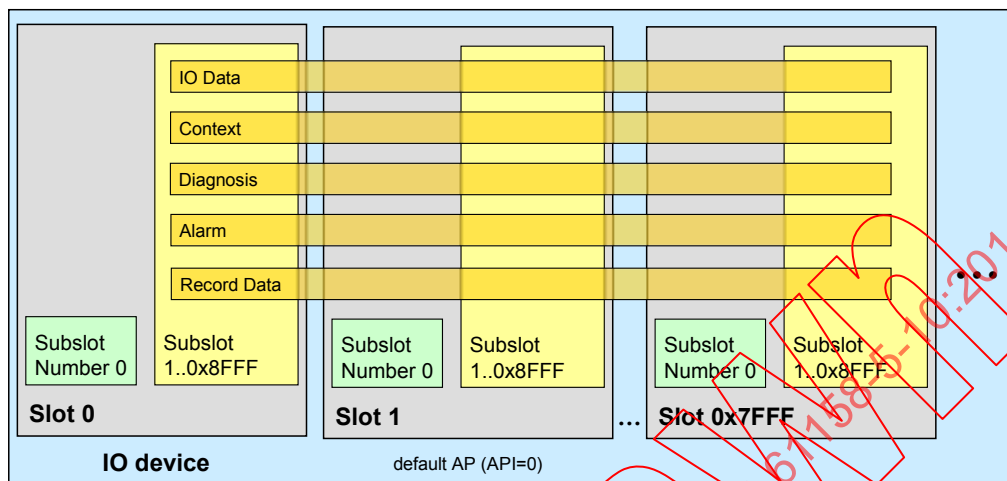


Légende

Anglais	Français
Alarm	Alarme
AP(API=x), $0 < x \leq 0xFFFFFFFF$, reserved for profile APs	AP(API=x), $0 < x \leq 0xFFFFFFFF$, réservé pour des AP de profils
AP (0<x<=0xFFFFFFFF)	AP, (0<x<=0xFFFFFFFF)
Associations	Associations
Context	Contexte
default AP, (API=0)	AP par défaut, (API=0)
Diagnosis	Diagnostics
IO controller	Appareil de commande E/S
IO Data	Données E/S
IO device	Appareil E/S
IO supervisor	Superviseur E/S
Isochronous Mode Application	Application en mode isochrone
Logbook Data	Données de journal
Physical Device Management	Gestion d'appareil physique
Record Data	Données d'enregistrement

Figure 17 – Vue d'ensemble de processus d'application

Un AP peut être distribué à plusieurs baies et sous-baies. La Figure 18 montre la relation entre les AP, les éléments de données, les baies et les sous-baies d'un appareil E/S. Les zones en gris illustrent le fait qu'il n'existe pas de «réelle» sous-baie 0. La sous-baie 0 ne doit pas non plus contenir de données E/S par exemple (voir Figure 18).



Légende

Anglais	Français
Alarm	Alarme
Context	Contexte
default AP, (API=0)	AP par défaut, (API=0)
Diagnosis	Diagnostic
IO Data	Données E/S
IO device	Appareil E/S
Record Data	Données d'enregistrement
Slot 0x7FFF	Baie 0x7FFF
Slot 1	Baie 1
Slot Number 0	Baie numéro 0
Subslot 1..0x8FFF	Sous-baie 1..0x8FFF

Figure 18 – Appareil E/S avec des AP, baies et sous-baies

La sous-baie Subslot 0x8000 à 0x9FFF doit exister uniquement dans l'API 0.

NOTE 1 La mise en correspondance d'un objet réel à un modèle d'adressage relève d'une initiative locale. Cela signifie qu'un même objet réel peut être mis en correspondance ou regroupé dans différents AP et sous-baies. Cependant, du point de vue du réseau, toutes les informations d'adresse API/baie/sous-baie sont uniques. Chaque API/baie/sous-baie peut donner une vue différente mais cohérente sur les objets visibles du réseau. Cependant, l'appareil E/S évite les conflits concernant l'utilisation multiple de données de sortie appelées "données de sortie en chevauchement". Une association d'applications issue d'un superviseur est toujours capable de prendre le contrôle sur des données de sortie et l'appareil de l'appareil de commande en est informé par une alarme spéciale.

Les règles suivantes doivent être appliquées pour l'AP:

- Les données d'identification d'instance d'appareil en tant que partie intégrante de l'ASE Context doivent toujours être identiques, quel que soit l'AP adressé.
- L'API 0 adresse le processus d'application
- Les API 1 à 0xFFFFFFFF sont réservés pour des profils E/S qui doivent être uniques.

Outre l'utilisation spécifique à un vendeur et à un profil du modèle d'adresses, les règles suivantes doivent s'appliquer pour l'indice (Index):

- Les indices 0 à 0x7FFF doivent être utilisés par le fabricant d'appareil pour les objets Record Data spécifiques à un utilisateur dans le domaine d'application de l'AP adressé. Pour l'API 0, les indices sont spécifiques à un vendeur. Pour les autres API, d'autres définitions spécifiées dans le profil connexe doivent être appliquées.

NOTE 2 Les indices 0 à 0xFF peuvent être utilisés pour la migration d'appareils conformes au CP3/1 de la CEI 61784-1 pour accéder aux mêmes objets Process Data avec le même indice.

- Les indices 0x8000 à 0xAFFF sont réservés à l'utilisation CEI 61158-6-10. Le domaine d'application de l'objet Record Data est une sous-baie particulière notamment dans le domaine d'application de l'AP adressé.
- Les indices 0xB000 à 0xBFFF sont réservés à l'utilisation du profil. Le domaine d'application de l'objet Record Data est une sous-baie particulière notamment dans le domaine d'application de l'AP adressé.
- Les indices 0xC000 à 0xCFFF sont réservés à l'utilisation CEI 61158-6-10. Le domaine d'application de l'objet Record Data est une baie particulière notamment dans le domaine d'application de l'AP adressé.
- Les indices 0xD000 à 0xDFFF sont réservés à l'utilisation du profil. Le domaine d'application de l'objet Record Data est une baie particulière notamment dans le domaine d'application de l'AP adressé.
- Les indices 0xE000 à 0xEBFF sont réservés à l'utilisation CEI 61158-6-10. Le domaine d'application de l'objet Record Data est une relation entre applications particulière.
- Les indices 0xEC00 à 0xEFFF sont réservés à l'utilisation du profil. Le domaine d'application de l'objet Record Data est une relation entre applications particulière.
- Les indices 0xF000 à 0xF3FF sont réservés à l'utilisation CEI 61158-6-10. Le domaine d'application de l'objet Record Data est un AP particulier.
- Les indices 0xF400 à 0xF7FF sont réservés à l'utilisation du profil. Le domaine d'application de l'objet Record Data est un AP particulier.
- Les indices 0xF800 à 0xFBFF sont réservés à l'utilisation CEI 61158-6-10. Le domaine d'application de l'objet Record Data est l'appareil.
- Les indices 0xFC00 à 0xFFFF sont réservés à l'utilisation du profil. Le domaine d'application de l'objet Record Data est l'appareil.

Outre l'utilisation spécifique à un vendeur et à un profil du modèle d'adresses, les règles suivantes doivent s'appliquer pour la baie (slot):

- Les baies 0 à 0x7FFF sont librement utilisables dans le domaine d'application de l'AP adressé. Pour l'API 0, les baies sont spécifiques à un vendeur. Pour les autres API, d'autres définitions spécifiées dans le profil connexe doivent être appliquées.
- Les baies 0x8000 à 0xFFFF sont réservées pour usage futur et ne doivent pas être utilisées, quel que soit l'AP.

Outre l'utilisation spécifique à un vendeur et à un profil du modèle d'adresses, les règles suivantes doivent s'appliquer pour la sous-baie (Subslot):

- La sous-baie 0x0000 est librement utilisable dans le domaine d'application de l'AP adressé, si `ARProperties.PullModuleAlarmAllowed` est utilisé.
- La sous-baie 0x0000 est utilisée pour adresser le module dans le domaine d'application de l'alarme «Pull Alarm» pour signaler le tirage d'un module, si `ARProperties.PullModuleAlarmAllowed` n'est pas utilisé.
- Les sous-baies 0x0001 à 0x7FFF sont librement utilisables dans le domaine d'application de l'AP adressé. Pour l'API 0, les sous-baies sont spécifiques à un vendeur. Pour les autres API, d'autres définitions spécifiées dans le profil connexe doivent être appliquées.
- Les sous-baies 0x8000 à 0x8FFF doivent être utilisées pour adresser les sous-modules d'interface et de port. Elles ne doivent être utilisées que dans l'API par défaut (zéro).

- Les sous-baies 0x9000 à 0x9FFF doivent être utilisées pour adresser les sous-modules d'interface et de port situés à droite ou au-dessus du nœud de transmission dans le cas spécial de redondance du système seulement. Elles ne doivent être utilisées que dans l'API par défaut (zéro).
- Les baies 0xA000 à 0xFFFF sont réservées pour usage futur et ne doivent pas être utilisées, quel que soit l'AP.

Chaque AP doit être adressé par un API (x) avec $0 < x \leq 0xFFFFFFFF$. L'utilisation d'AP supplémentaires doit être limitée aux définitions officielles de profils.

7.1.6.2 Élément de service application

Un élément de service application (ASE), tel que défini dans l'ISO/CEI 9545, est un jeu de fonctions d'application qui fournissent un moyen pour l'interfonctionnement des processus d'application dans un but spécifique. Les ASE fournissent un jeu de services pour acheminer les demandes et réponses à destination et en provenance de processus d'application et de leurs objets.

La couche application propose les ASE suivants:

ASE IO Data

L'ASE IO Data fournit un jeu de services pour acheminer des données E/S de façon cyclique. Ces données appartiennent toujours aux baies/sous-baies qui ont été configurées en termes de l'ASE Context. Les objets IO Data contiennent un statut pour émission. Facultativement, l'ASE IO Data offre la possibilité de partager les données d'entrées d'un appareil E/S avec d'autres appareils E/S. Les données E/S peuvent également être lues et écrites par un superviseur E/S de façon acyclique.

ASE Record Data

L'ASE Record Data fournit un jeu de services pour acheminer des données de façon acyclique. L'application de l'appareil de commande E/S ou du superviseur E/S demande chaque émission sur une base individuelle. L'ASE Record Data peut être relatif à tous les AP d'un appareil E/S.

ASE LogBook Data

L'ASE Logbook Data fournit un jeu de services pour lire des données de journaux. L'application de l'appareil de commande E/S ou du superviseur E/S demande chaque émission sur une base individuelle. L'ASE Logbook Data peut être relatif à tous les AP d'un appareil E/S.

ASE Diagnosis

L'ASE Diagnosis fournit des services pour l'appareil de commande E/S ou le superviseur E/S pour lire des informations de diagnostic (Diagnosis information) issues d'un appareil E/S.

ASE Alarm

L'ASE Alarm fournit un jeu de services pour acheminer des alarmes émises par l'appareil E/S ou l'appareil de commande E/S. L'appareil de commande E/S ou l'appareil E/S assigné acquitte l'alarme.

ASE Context

L'ASE Context fournit un jeu de services pour

- acheminer les paramètres et la configuration de l'appareil selon la description de l'appareil
- identifier des points d'extrémités d'AR
- maintenir des paramètres d'AR et de CR (temporisations, modes, ..)
- et établir ou libérer une association entre des AP individuels.

ASE Isochronous Mode Application

L'ASE Isochronous Mode Application fournit un jeu de services pour paramétrer et synchroniser des processus d'application isochrones.

ASE Application Relationship

L'ASE Application Relationship (ASE AR) fournit un modèle de description pour les types AR distincts. Cela inclut leurs caractéristiques de transfert ainsi que leurs états de communication courants.

ASE Physical Device Management

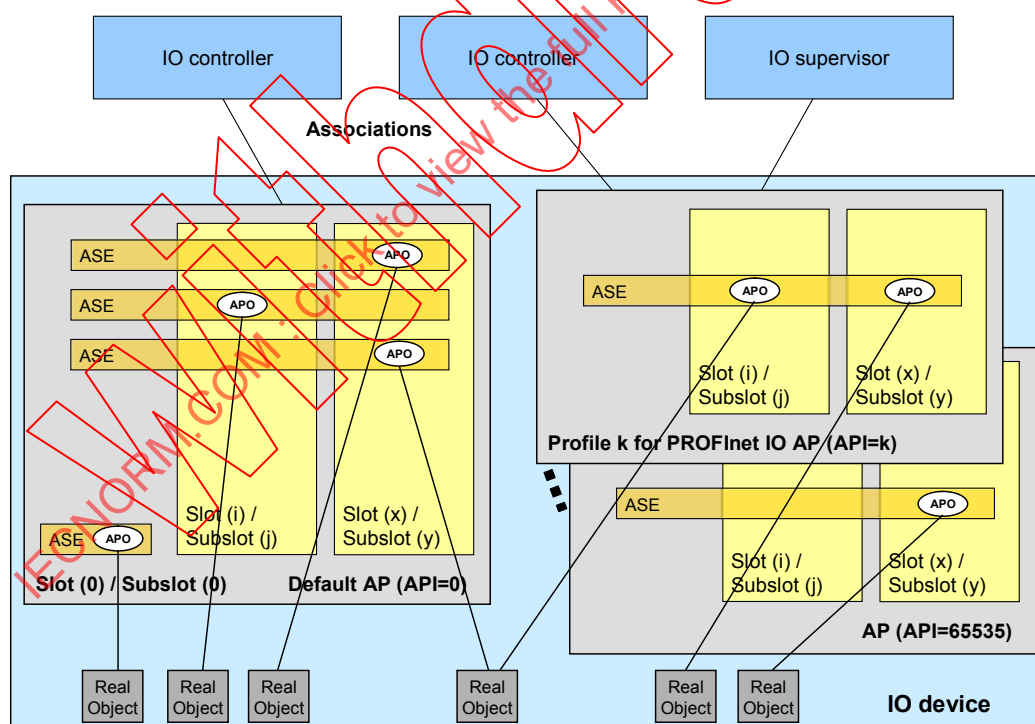
L'ASE Physical Device Management fournit un jeu de services pour gérer le démarrage de l'appareil physique. Il inclut l'affectation d'un Station Name (nom de station) et d'autres paramètres d'adresse IP.

NOTE Le terme «physical device» (appareil physique) sous-entend qu'il y a une seule instance de cet ASE pour un appareil d'automation «réel», quel que soit le nombre d'instances de types d'appareil dans l'appareil d'automation. De toute façon, il n'y aura qu'une seule instance pour la plupart des appareils.

7.1.6.3 Objets de processus d'application

Un objet de processus d'application (APO pour «application process object») est une représentation réseau d'un aspect spécifique d'un processus d'application (AP pour «application process»). Chaque APO représente un jeu d'informations et de moyens de traitements d'un AP qui sont accessibles par le biais de services de la couche application. Les APO sont utilisés pour représenter ces moyens à d'autres AP dans un système.

Afin de permettre à un AP de communiquer avec un AP d'un autre service, des APO doivent être disponibles. Les objets de processus d'application (objets virtuels) représentent des objets de processus existants (objets réels) qu'un processus d'application a rendus visibles et accessibles à la communication (voir Figure 19).



Légende

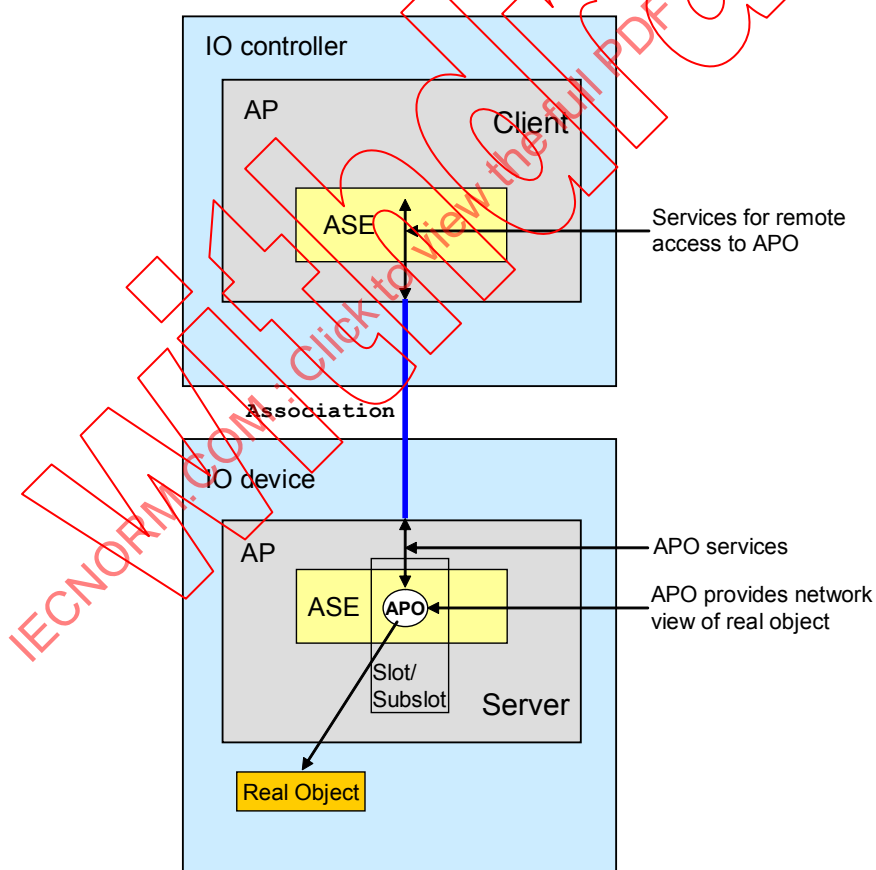
Anglais	Français
AP (API = 65535)	AP (API = 65535)
APO	APO
ASE	ASE
Associations	Associations

Anglais	Français
default AP(API = 0)	AP par défaut, (API=0)
IO controller	Appareil de commande E/S
IO device	Appareil E/S
IO supervisor	Superviseur E/S
Profile k for PROFINet IO AP (API = k)	Profil k pour AP E/S PROFINet IO AP (API = k)
Real Object	Objet réel
Slot(0)/Subslot(0)	Slot(0)/Subslot(0)
Slot(i)/Subslot(j)	Slot(i)/Subslot(j)
Slot(x)/Subslot(y)	Slot(x)/Subslot(y)

Figure 19 – Processus d'application avec objets de processus d'application (APO)

Dans la Figure 20, un AP distant agissant comme Client peut accéder à l'objet réel en envoyant des demandes par l'intermédiaire de l'APO qui représente l'objet réel. Les aspects locaux de l'AP assurent la conversion entre la vue réseau (l'APO) de l'objet réel et la vue d'AP interne de l'objet réel.

Au sein d'un AP, un APO est identifié par une baie, une sous-baie et un indice. L'espace adresse qui est défini par la baie, la sous-baie et l'indice peut être utilisé par plusieurs AP.



Légende

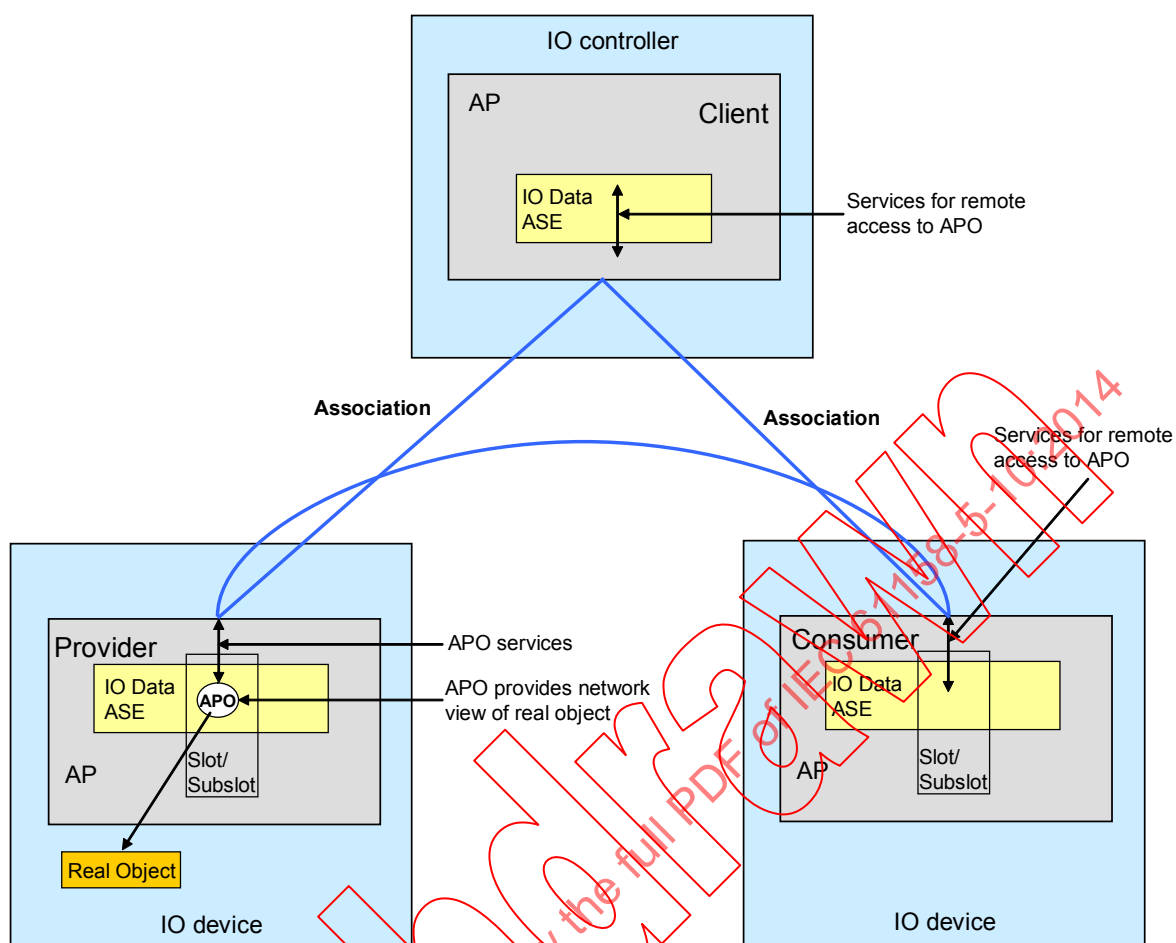
Anglais	Français
AP	AP
APO provides network view of real object	L'APO donne une vue réseau de l'objet réel

Anglais	Français
APO services	Services APO
ASE	ASE
APO	APO
Association	Association
Client	Client
IO controller	Appareil de commande E/S
IO device	Appareil E/S
Real Object	Objet réel
Server	Serveur
Services for remote access to APO	Services pour accès distant à l'APO
Slot/Subslot	Baie/Sous-baie

Figure 20 – Accès à un APO distant

La Figure 21 montre une association Client/Server (Client/Serveur) et une association Provider/Consumer (Fournisseur/Consommateur). Le Client peut accéder à l'objet réel en envoyant des demandes par l'intermédiaire de l'APO qui représente l'objet réel. Le Consommateur peut s'abonner à tout ou partie de l'APO distant. L'association Fournisseur/Consommateur est toujours relative à l'ASE IO Data. Les aspects locaux de l'AP assurent la conversion entre la vue réseau (l'APO) de l'objet réel et la vue d'AP interne de l'objet réel.

NOTE La mise en correspondance de données réelles à des APO visibles du réseau relève d'une initiative locale. Cependant, le vendeur d'appareil doit fournir des moyens locaux de prévenir les conflits avec des sorties en chevauchement et des variables inscriptibles.



Légende

Anglais	Français
AP	AP
APO	APO
APO provides network view of real object	L'APO donne une vue réseau de l'objet réel
APO services	Services APO
ASE	ASE
Association	Association
Client	Client
Consumer	Consommateur
IO controller	Appareil de commande E/S
IO Data	Données E/S
IO device	Appareil E/S
Provider	Fournisseur
Real Object	Objet réel
Services for remote access to APO	Services pour accès distant à l'APO
Slot/Subslot	Baie/Sous-baie

Figure 21 – Accès à un APO distant pour association fournisseur/consommateur

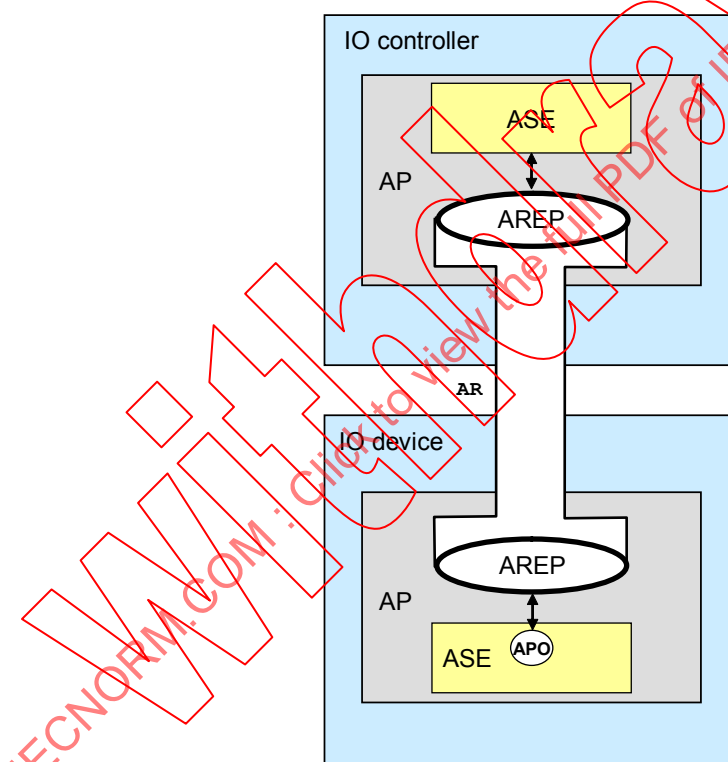
7.1.7 Relation entre applications

7.1.7.1 Généralités

Une Relation d'applications (AR) est une relation coopérative entre deux ou plusieurs AP dans le but d'échanger des informations et de coordonner leur fonctionnement conjoint (voir Figure 22). La relation est activée par l'échange d'unités de données de protocole application (APDU). La couche application utilise différents types d'AR qui se distinguent par leur caractéristique d'acheminement.

7.1.7.2 Point d'extrémité de relation entre applications

Un AP a accès à la communication en utilisant des points d'extrémité de relation entre applications (AREP, voir l'ISO 7498-1). Un ou plusieurs AREP sont fixes et assignés de manière univoque à un AP. Ces points d'extrémité sont adressés par l'AP par l'intermédiaire d'un identificateur (AREP ID). Ces identificateurs sont spécifiques à un appareil et ne sont pas définis par la communication elle-même. Les AR sont définies comme un jeu d'AREP coopératifs. Entre deux AP, il peut exister une ou plusieurs AR, chacune ayant des AREP uniques. La Figure 22 montre un exemple d'une AR avec deux AREP.



Légende

Anglais	Français
IO device	Appareil E/S
IO controller	Appareil de commande E/S
AP	AP
ASE	ASE
APO	APO
AR	AR
AREP	AREP

Figure 22 – Exemple d'une AR avec deux AREP

7.1.7.3 Vue d'ensemble de relations entre applications

La couche application propose les types d'AR suivants:

NOTE L'AR peut être identifiée avec un UUID unique appelé «AR UUID» (UUID d'AR).

IO AR: relation entre applications entre le processus d'application d'un appareil de commande E/S et le Processus d'application connexe d'un appareil E/S pour les besoins suivants:

- échange cyclique de données d'entrée avec l'appareil de commande E/S au moyen d'une Input CR (BBUU CR)
- échange cyclique de données de sortie avec l'appareil de commande E/S au moyen d'une Output CR (BBUU CR)
- échange cyclique (en multidiffusion) des données d'entrée ou des données de sortie au moyen d'une M CR (BBUU-OM CR)
- transfert acyclique de données pour, par exemple, paramétrisation, configuration, données E/S et diagnostic (objets Record Data) au moyen d'une RDCM CR (QQCB-CO CR)
- émission de données d'alarme dans les deux sens au moyen d'une Alarm CR (QQCB-CO CR)

L'IO AR doit seulement être utilisée conjointement aux API 0 à 0xFFFFFFFF.

Supervisor AR: relation entre applications entre:

- les processus d'application d'un superviseur E/S et un appareil E/S,
 - les processus d'application d'un serveur de paramètres E/S et un appareil E/S connexe,
 - les processus d'application d'un appareil de commande E/S et un appareil E/S,
- pour les besoins suivants:
- mêmes fonctions que celles fournies par l'IO AR avec les API 0 à 0xFFFFFFFF,
 - prise en charge supplémentaire d'une IO AR (commandée par le superviseur)

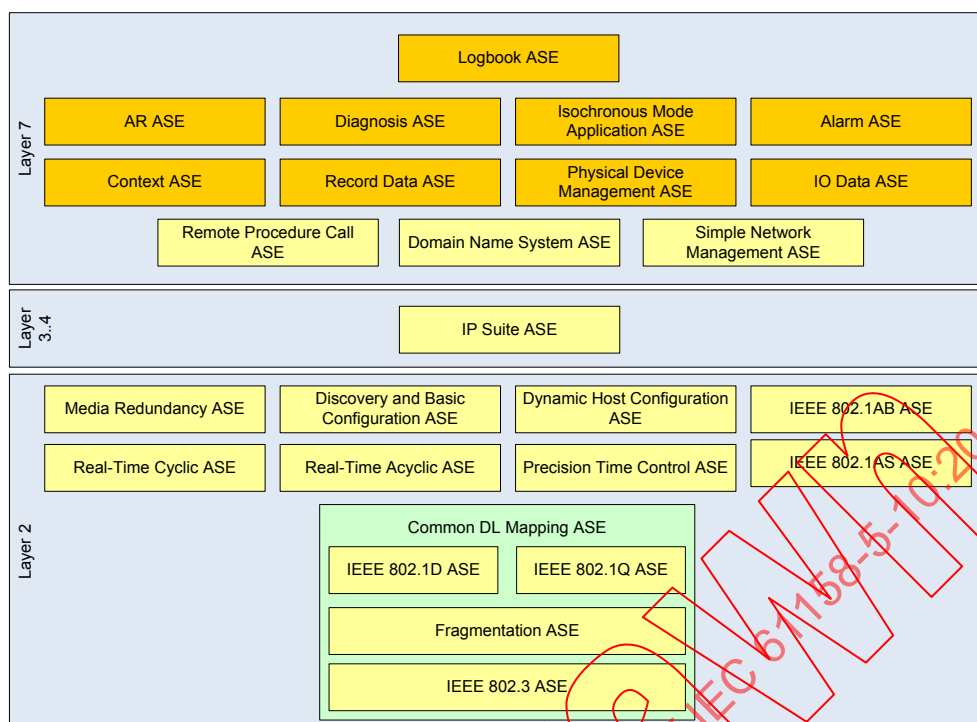
Implicit AR: (AR UUID = UUID_NIL) relation entre applications entre:

- les processus d'application d'appareil de commande E/S, d'appareils E/S, de superviseur E/S à des valeurs de Read de l'objet ASE».

L'Implicit AR peut être utilisée conjointement à tous les API.

7.1.8 ASE de la FAL

La Figure 23 montre les ASE FAL périphériques décentralisées en fonction de leur couche OSI, y compris les services communs ASE FAL.



Légende

Anglais	Français
Logbook ASE	ASE "Logbook" (Journal de bord)
AR ASE	ASE AR (Relation en applications)
Diagnosis ASE	ASE Diagnosis (Diagnostic)
Isochronous Mode Application ASE	ASE Isochronous Mode Application (Application mode isochrone)
Alarm ASE	ASE Alarm
Context ASE	ASE Context
Record Data ASE	ASE Record Data
Physical Device Management ASE	ASE Physical Device Management
IO Data ASE	ASE IO Data
Remote Procedure Call ASE	ASE Remote Procedure Call
Domain Name System ASE	ASE Domaine Name System
Simple Network Management ASE	ASE Simple Network Management
Layer 7	Couche 7
IP Suite ASE	ASE IP Suite
Layer 3.4	Couche 3.4
Media Redundancy ASE	ASE Media Redundancy
Discovery and Basic Configuration ASE	ASE Discovery and Basic Configuration
Dynamic Host Configuration ASE	ASE Dynamic Host Configuration
Real Time Cyclic ASE	ASE Real Time Cyclic
Real Time Acyclic ASE	ASE Real Time Acyclic
Precision Time Control ASE	ASE Precision Time Control
IEEE 802 1AB ASE	ASE IEEE 802 1AB
Common DL Mapping ASE	ASE Common DL Mapping
IEEE 802 1D ASE	ASE IEEE 802 1D

Anglais	Français
IEEE 802.1Q ASE	ASE IEEE 802.1Q
Fragmentation ASE	ASE Fragmentation
IEEE 802.3 ASE	ASE IEEE 802.3
Layer 2	Couche 2

Figure 23 – Architecture de communication des ASE de FAL

7.2 Types de données des ASE

Les types de données définis à l'Article 5 sont pris en charge.

7.3 ASE

7.3.1 ASE Record Data

7.3.1.1 Vue d'ensemble

Dans l'environnement de couche application, les processus d'application contiennent des données que des applications distantes sont capables de lire et écrire. L'ASE Record Data définit des attributs des objets Record Data d'usage général et fournit un jeu de services utilisés pour lire et écrire leurs valeurs. En outre, le client peut annuler un service demandé en cours en raison de conditions spéciales d'application telles qu'une reconfiguration dynamique.

Un objet Record Data particulier peut être lu et écrit, partiellement ou complètement (si pris en charge par l'application). La connexion locale entre un objet Record Data et un objet physique ou du monde réel ne s'inscrit pas le domaine d'application de cette définition.

NOTE Un objet Record Data peut être connecté à un même objet du monde réel pour lire et écrire ou il peut être affecté à un objet réel pour lire la valeur et à un autre pour écrire la valeur.

L'ASE de Record Data permet l'adressage des objets Record Data liés à des couches spécifiques du modèle d'adressage, tel que décrit en 7.1.6.1. Le domaine d'application d'un objet d'enregistrement de données particulières est dérivé de l'indice comme indiqué en 7.1.6.1.

Des objets Record Data appartenant

- à une sous-baie au sein d'un AP sont adressés par l'API, le numéro d'emplacement, numéro de sous-baie, et l'indice.
- à une baie au sein d'un AP sont adressés par l'API, le numéro d'emplacement et l'indice.
- à un AP sont adressés par API.
- à une relation entre applications sont adressés par l'ARUUID et l'indice ou s'il est utilisé une AR implicite par le Target AR UUID et l'Index
- à un appareil sont adressés uniquement par Index.

L'appareil de commande E/S et le superviseur E/S sont capables de lire ou d'écrire la valeur d'un objet Record Data d'un appareil E/S. En outre, un appareil E/S est capable de lire ou d'écrire la valeur d'un objet Record Data d'un serveur de paramètres E/S. Un superviseur E/S est capable de lire ou d'écrire la valeur d'un objet Record Data d'un appareil de commande E/S. En utilisant l'attribut Device Access, un superviseur E/S est capable de lire ou d'écrire tous les objets Record Data si cela est autorisé par le processus d'application de l'appareil E/S.

Le processus d'application client peut utiliser l'ASE Context pour établir une association afin d'avoir accès aux objets Record Data d'un serveur. L'accès en lecture est également possible sans connexion par l'intermédiaire de l'Implicit AR.

L'accès à des objets Record Data est réalisé conformément au modèle d'accès Client/Serveur. Le modèle Client/Serveur est caractérisé en ce que l'application Client envoie une demande de lecture ou d'écriture à une application Serveur qui répond en conséquence.

Le modèle formel de l'ASE Record Data est présenté par la spécification pour la classe Record Data, contenant une description de ses attributs, de ses services ainsi que de ses invocations, suivie d'une spécification particulière des services.

7.3.1.2 Liaison des services Record Data

Les services de l'ASE Record Data sont liés à aux services de niveau inférieur suivants, (voir Tableau 107):

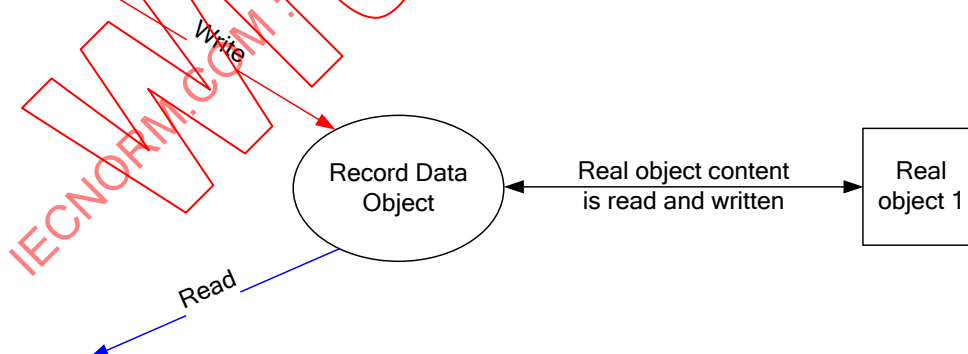
Tableau 106 – Liaison des services Record Data

Service	Liaison	Usage
Read	lié à RPC::Read	utilisé par l'appareil de commande E/S ou le superviseur E/S
Read Query	lié à RPC::Read	utilisé par l'appareil de commande E/S ou le superviseur E/S
Write	lié à RPC::Write	utilisé par l'appareil de commande E/S ou le superviseur E/S

7.3.1.3 Spécification de la classe Record data (données d'enregistrement)

7.3.1.3.1 Modèle

Un objet Record Data peut être connexe à un objet réel comme montré à la Figure 24.

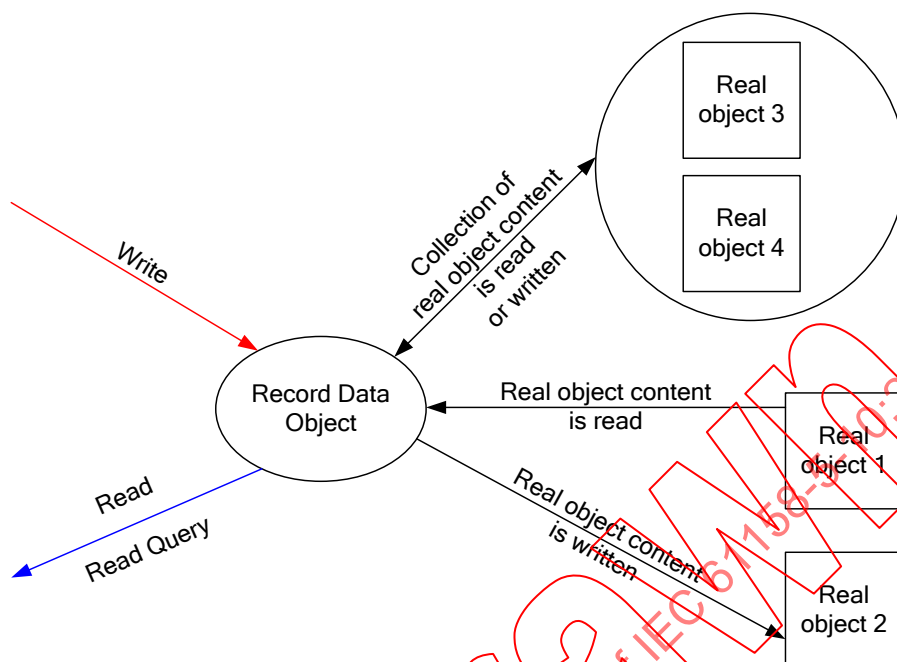


Légende

Anglais	Français
Write	Write (Ecrire)
Read	Read (Lire)
Record Data Object	Objet de données d'enregistrement
Real object content is read and written	Le contenu de l'objet réel est lu et écrit.
Real object 1	Objet réel 1

Figure 24 – Relation d'un objet Record Data à un seul objet réel

Un objet Record Data peut être relié à un ou plusieurs objets réels (voir Figure 25).



Légende

Anglais	Français
Write	Ecrire
Read	Lire
Record Data Object	Objet de données d'enregistrement
Real object content is read and written	Le contenu de l'objet réel est lu et écrit.
Real object 1	Objet réel 1
Real object 2	Objet réel 2
Real object 3	Objet réel 3
Real object 4	Objet réel 4
Read Query	Interrogation de lecture
Collection of real object content is read and written	Le groupement du contenu de l'objet réel est lu et écrit.
Real object content is read	Le contenu de l'objet réel est lu.
Real object content is written	Le contenu de l'objet réel est écrit.

Figure 25 – Relation d'un objet Record Data à deux objets réels

Un objet Record Data est décrit par le modèle suivant:

ASE: ASE Record Data

CLASS: Record Data

CLASS ID: pas utilisé

PARENT CLASS: TOP

ATTRIBUTES:

- 1 (m) Attribut-clé: Identificateur
- 1.1 (m) Attribut: API
- 1.2 (m) Attribut: Slot Number
- 1.3 (m) Attribut: Subslot Number
- 1.4 (m) Attribut: Index

2	(m) Attribut:	Read Record Data Description
3	(m) Attribut:	Read Partial Access
4	(m) Attribut:	Write Record Data Description
5	(m) Attribut:	Write Partial Access
6	(m) Attribut:	Shared Write Access
7	(o) Attribut:	Write Persistence Flag

SERVICES:

1	(m) OpsService:	Read
2	(o) OpsService:	Read Query
3	(m) OpsService:	Write
4	(m) OpsService:	Write Combined Object Container
5	(m) OpsService:	Local Write Multiple
6	(m) OpsService:	Local New Write Multiple

7.3.1.3.2 Attributs**Identifier**

Cet attribut clé est composé d'attributs qui définissent le contexte auquel l'objet Record Data appartient. Cet «Identifier» (identificateur) est unique dans l'appareil EIS et ne doit pas être utilisé par un autre objet.

API

Cet attribut définit le processus d'application auquel appartiennent les Input Data Elements.

Type d'attribut: Unsigned32

Valeurs permises: 0 comme processus d'application par défaut. Les valeurs 1 à 0xFFFFFFFF sont réservées pour usage futur dans les lignes directrices des profils.

Slot Number

Cet attribut définit le module auquel appartient l'objet Record Data.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient l'objet Record Data.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Indice

Cet attribut définit l'indice de l'objet Record Data.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0xFFFF

Read Record Data Description

Cet attribut peut contenir soit une Simple Data Description, soit une Array Data Description, soit une Record Data Description. Pour la description de données, le modèle utilisé est celui donné dans la CEI 61158–1.

Read Partial Access

Cet attribut définit si un accès partiel en lecture à l'objet Record Data est pris en charge ou non. Cet attribut commande le comportement du service Read. "Partial access égal à TRUE" signifie que le contenu de l'objet Record Data peut être lu du premier octet jusqu'à l'octet défini par la longueur donnée dans la demande de service, et ce, même si la longueur est plus courte que la longueur de l'objet Record Data. Dans tous les cas, l'objet tout entier sera lu si la longueur donnée dans le service est supérieure ou égale à la longueur de l'objet. "Partial access égal à FALSE" signifie que le contenu de l'objet Record Data peut seulement être lu en entier. Cela implique que la longueur donnée dans le service doit être supérieure ou égale à la longueur de l'objet. Autrement, il doit retourner une réponse négative avec le code d'erreur "Invalid type".

Type d'attribut: Boolean

Write Record Data Description

Cet attribut peut contenir soit une Simple Data Description, soit une Array Data Description, soit une Record Data Description. Pour la description de données, le modèle utilisé est celui donné dans la CEI 61158–1.

Write Partial Access

Cet attribut définit si un accès partiel en écriture à l'objet Record Data est pris en charge ou non. Cet attribut commande le comportement du service Write. "Partial access égal à TRUE" signifie que le contenu de l'objet Record Data doit être écrit du premier octet jusqu'à l'octet défini par la longueur donnée dans la demande de service, et ce, même si la longueur est plus courte que la longueur de l'objet Record Data. Dans tous les cas, l'objet tout entier sera écrit si la longueur donnée dans le service est égale à la longueur de l'objet. "Partial access égal à FALSE" signifie que le contenu de l'objet Record Data peut seulement être écrit en entier. Cela implique que la longueur donnée dans le service doit être égale à la longueur de l'objet. Autrement, il doit retourner une réponse négative avec le code d'erreur "Write Length Error". Si la longueur dans le service Write dépasse la longueur de l'objet Record Data, une réponse négative avec le code d'erreur "Write Length Error" doit être retournée.

Type d'attribut: Boolean

Shared Write Access

Cet attribut définit si, oui ou non, un accès en écriture à l'objet Record Data est pris en charge pour le client connecté par le biais d'une relation d'applications partagée. Cet attribut commande le comportement du service Write. "Shared Write Access égal à TRUE" signifie que le contenu de l'objet Record Data peut être écrit à partir du client connecté par le biais d'une relation d'applications partagée. Dans ce cas, l'application locale est responsable de la cohérence des données. "Shared Write Access égal à FALSE" signifie que le contenu de l'objet Record Data doit être écrit à partir du client connecté par le biais d'une relation d'applications partagée.

Type d'attribut: Boolean

Write Persistence Flag

Cet attribut facultatif doit définir le comportement en persistance pour la valeur de l'objet Record Data au cours d'un cycle de puissance. Il doit être stocké de manière non volatile si la valeur est mise à TRUE. Autrement, la valeur écrite est volatile et, après un cycle de puissance, absente.

NOTE 1 Cet attribut et une propriété locale d'un appareil mise en œuvre au nom du fournisseur d'appareil.

Le processus d'application doit mettre en œuvre le comportement de stockage persistance du Tableau 107 selon cet attribut et le paramètre Prm Flag de service d'indication Write.

Tableau 107 – Comportement de la persistance des objets Record Data

Write Persistence Flag	Write.ind (Prm Flag)	Stockage de persistance
FALSE (faux)	FALSE (faux)	doit être stocké de manière volatile
FALSE (faux)	TRUE (vrai)	doit être stocké de manière volatile
TRUE (vrai)	FALSE (faux)	doit être stocké de manière non volatile avant Write.rsp(+)
TRUE (vrai)	TRUE (vrai)	Write.ind: doit seulement être stocké temporairement de manière volatile et doit répondre immédiatement Write.rsp(+) avant de recevoir l'indication "Prm End" Application Ready.req: Avant d'émettre la demande Application Ready, toutes les données intermédiaires stockées doivent faire l'objet d'une vérification de cohérence, seul le jeu de données cohérentes doit être stocké de manière non volatile, doit encore faire l'objet d'une vérification de cohérence dans la mémoire non volatile et envoyer la demande Application Ready par la suite.
NOTE Seuls des jeux cohérents de données au cours de la phase de paramétrisation sont stockés de manière non volatile. Une défaillance d'AR au cours de cette phase n'altère pas les données d'enregistrement stockées antérieurement et le tampon intermédiaire est effacé.		

NOTE 2 Un processus d'application peut éviter la sauvegarde en persistance de valeurs inchangées par des moyens locaux, par exemple en utilisant des sommes de contrôle.

Type d'attribut: Boolean

7.3.1.3.3 Invocation de l'objet Record Data

Pour l'invocation de l'objet Record Data, les règles suivantes s'appliquent:

- Les objets Record Data ne doivent pas dépasser la longueur totale de $2^{32}-65$ octets. L'attribut Record Data Description doit être positionné en conséquence.
- Au moins un service doit être autorisé pour l'accès à l'objet.
- Au moins une AR doit avoir accès à l'objet.
- Les droits d'accès doivent être établis en conséquence au(x) service(s) autorisé(s).
- Une demande de service d'enregistrement de données (à savoir lecture, interrogation de lecture, écriture ou écriture multiple) doit recevoir une réponse en moins de 300 s; autrement, le client du service peut supposer que le service a expiré.

7.3.1.3.4 Focalisation d'un objet Record Data

Les informations d'un objet Record Data sont filtrées selon les règles suivantes:

- Si l'objet Record Data est adressé sur l'AR établi ou sur une lecture implicite avec Target AR UUID, l'information
 - est filtrée selon les données d'identification **prévues** (voir ASE Context), si l'objet est adressé ne fait pas partie de la relation entre applications adressée.

NOTE 1 Par exemple, en obtenant des informations de diagnostic à partir d'une baie faisant partie des données d'identification prévue, seul le diagnostic lié à des sous-modules dans les baies adressées, qui font partie des données d'identification prévues, est renvoyé.
 - est filtrée selon les données d'identification **réelles** (voir ASE Context), si l'objet est adressé ne fait pas partie de la relation entre applications adressée.

NOTE 2 Par exemple, en obtenant des informations de diagnostic à partir d'une baie ne faisant pas partie des données d'identification prévue, le diagnostic lié à **tous** les sous-modules dans les baies adressées, qui font partie des données d'identification réelles, est renvoyé.
- Si l'objet Record Data est adressé sur l'AR implicite ou sur une lecture implicite sans préciser de Target AR UUID, l'information est toujours filtrée en fonction des données d'identification **réelles** (voir ASE Context).

- Si l'objet Record Data est sur la couche de l'appareil, l'information est toujours filtrée en fonction des données d'identification **réelles** (voir ASE Context).

La raison de ces règles de filtrage est que l'utilisateur ne veut voir que les objets qui l'intéressent.

7.3.1.3.5 Hiérarchie des codes d'erreur de l'objet Record Data

Le PNIO Status (statut PNIO) des services Record Data doit être défini en fonction de l'erreur courante et doit être constitué des éléments Error code, Error Decode, Error code 1, et Error code 2.

Si PNIO Status est mis sur (0x00, 0x00, 0x00, 0x00), le cas spécial «No error» s'applique.

Si une erreur doit être renvoyée, Error code est mis à IODReadRes lors de l'utilisation des services Read ou Read Query, ou à IODWriteRes lors de l'utilisation du service Write.

Une erreur de protocole est indiquée par Error decode = PNIO; pour plus de détails concernant Error code 1 et Error code 2 voir la CEI 61158-6-10.

Une erreur d'application sur l'accès à un objet Record Data est indiquée par Error decode = PNIORW. Dans ce cas, la hiérarchie de génération des codes d'erreur suivants s'applique pour Error code 1. Error code 2 peut être mis à une valeur spécifique à l'utilisateur.

- a) Si un objet Record Data est lu et un UUID d'AR cible a été spécifié (donc il n'est pas NIL) lorsque l'AR voulue n'existe pas:
Error code 1 est mis à «access / invalid parameter»; ce qui signifie que l'AR cible désirée n'existe pas.
- b) Si un objet Record Data est écrit lorsque l'état AR est mis à BACKUP:
Error code 1 est mis à «access / backup»; ce qui signifie que l'écriture d'enregistrements dans l'état BACKUP de l'AR n'est pas autorisée.
- c) Si un objet Record Data est accessible (lecture ou écriture), lorsque l'API n'est pas pris en charge ou est non valide:
Error code 1 est mis à «access / invalid area»; ce qui signifie que l'API désiré n'existe pas.
- d) Si un objet Record Data est accessible (lecture ou écriture), lorsque la baie n'existe pas ou est vide:
Error code 1 est mis à «access / invalid slot»; ce qui signifie que l'API désiré existe, mais que la baie désirée n'existe pas ou est vide.
- e) Si un objet Record Data est accessible (lecture ou écriture), lorsque la sous-baie n'existe pas ou est vide:
Error code 1 est mis sur «access / invalid subslot»; ce qui signifie que l'API désiré et la baie existent, mais que la sous-baie désirée n'existe pas ou est vide.
- f) Si un objet Record Data est accessible (lecture ou écriture), lorsque l'indice n'est pas pris en charge:
Error code 1 est mis à «access / invalid index»; ce qui signifie que l'API, la baie, et la sous-baie désirés existent, mais que l'indice souhaité n'est pas pris en charge.
- g) Si un objet Record Data est écrit, lorsque l'accès en écriture n'est pas pris en charge:
Error code 1 est mis sur «application / write error», «access / type conflict», «access / state conflict» ou «access / access denied».
- h) Si un objet Record Data est lu, lorsque l'accès en lecture n'est pas pris en charge:
Error code 1 est mis sur «application / read error», «access / type conflict», «access / state conflict» ou «access / access denied».
- i) Erreurs de contenu dans la structure de l'objet Record Data à écrire:

Error code 1 et mis sur «access / write length error», «access / invalid range», «access / invalid parameter» ou «access / invalid type».

j) Erreurs de ressources lors de l'accès sur l'objet Record Data:

Error code 1 et mis sur «resource / read constrain conflict», «resource / write constrain conflict», «resource / resource busy», «resource / resource unavailable», ou «resource / user specific 12» sur «resource / user specific 19».

Cela indique une erreur temporaire et peut conduire à un accès réussi lors des tentatives des clients.

k) Erreurs d'application lors de l'accès sur l'objet Record Data:

Error code 1 et mis sur «application / module failure», «application / busy», «application / version conflict», «application / feature not supported», ou «application / user specific 1» sur «application / user specific 6».

7.3.1.4 Spécification des services de Record data

7.3.1.4.1 Services de lecture

7.3.1.4.1.1 Read

Ce service confirmé peut être utilisé pour lire la valeur d'un objet Record Data. Ce service doit être utilisé seulement conjointement à l'AR implicite (Implicit AR), IO AR ou Supervisor AR. En plus de ce service de lecture commun, des services complémentaires sont définis pour lire des données d'application spécifiques. Tous ces services de lecture utilisent les mêmes paramètres, seul le champ de données est différent. Le Tableau 108 indique les paramètres du service. Ce service peut également être utilisé pour lire les données d'un utilisateur et d'un profil supplémentaire.

L'attribut Read Partial Access commande le fonctionnement de la réponse lorsque la longueur demandée est plus courte que la longueur de l'objet Record Data (voir la description des attributs).

Tableau 108 – Read

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
API	M	M(=)		
Target AR UUID	U	U(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Index	M	M(=)		
Seq Number	M	M(=)		
Length	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Longueur			M	M(=)
Data			M	M(=)
Result (-)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)

Nom de paramètre	req	ind	rsp	cnf
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

API

Si le domaine d'application de l'objet Record Data est une sous-baie, une baie ou un API, ce paramètre doit être utilisé pour adresser l'AP désiré.

Target AR UUID

Si le domaine d'application de l'objet Record Data est la relation entre applications, ce paramètre doit être utilisé conjointement à l'AR implicite (Implicit AR) pour adresser l'AR souhaitée au sein de l'appareil E/S.

Slot Number

Si le domaine d'application de l'objet Record Data est une sous-baie ou une baie, ce paramètre doit être utilisé pour adresser la baie désirée (habituellement un module).

Subslot Number

Si le domaine d'application de l'objet Record Data est une sous-baie, ce paramètre doit être utilisé pour adresser la sous-baie désirée (habituellement un sous-module).

Indice

Le paramètre Index doit être utilisé pour adresser l'objet Record Data désiré. Il est en outre utilisé pour déduire le domaine d'application d'un objet Record Data.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16}-1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de Seq Number doit être incrémenté pour chaque demande de service au cours de la session.

NOTE 1 Si, par exemple, une demande de service d'écriture a le Seq Number mis à «17», une demande de service de lecture suivante doit avoir le Seq Number mis à «18».

En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Une confirmation avec un Seq Number qui n'est pas en cours sera ignorée. Une indication avec un Seq Number qui n'est pas en cours sera rejetée. Le Seq Number doit être maintenu séparément pour chaque AR établie.

Length

Le paramètre Length indique le nombre d'octets d'un objet Input Data qui doit être lu. La longueur permise se situe dans la plage 2^0 à $2^{32}-256$. Si la longueur demandée dépasse la longueur réelle de l'élément d'objet Input Data, la longueur réelle avec des données doit être donnée en réponse. Si la longueur demandée est plus courte que la longueur de l'élément d'objet Input Data, une erreur de paramètre doit être le résultat.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Length

Le paramètre Length (longueur) indique le nombre d'octets de l'objet Data qui a été lu.

Data

Le paramètre Data contient la valeur de l'objet spécifique qui a été lue et est constitué du nombre d'octets indiqués dans le paramètre Length. Ce paramètre doit être constitué des types de données définis à l'Article 5. Il existe une signification particulière pour les différents services de ces données, elle est décrite dans les parties concernant les services spéciaux. Le Tableau 109 indique les paramètres des services de lecture spéciaux.

Tableau 109 – Services Read (lecture)

Nom de service	Paragraphe
Read Output Data	7.3.2.5.16
Read Output Substitute Data	7.3.2.5.17
Read Input Data	7.3.2.5.9
Read LogBook Data	7.3.3.3.1
Read Diagnosis in channel coding for one subslot	7.3.4.3.5
Read Diagnosis in all codings for one subslot	7.3.4.3.6
Read Maintenance required in channel coding for one subslot	7.3.4.3.7
Read Maintenance required in all codings for one subslot	7.3.4.3.8
Read Maintenance demanded in channel coding for one subslot	7.3.4.3.9
Read Maintenance demanded in all codings for one subslot	7.3.4.3.10
Read Diagnosis, Maintenance, Qualified and Status for one subslot	7.3.4.3.11
Read Diagnosis in channel coding for one slot	7.3.4.3.12
Read Diagnosis in all codings for one slot	7.3.4.3.13
Read Maintenance required in channel coding for one slot	7.3.4.3.14
Read Maintenance required in all codings for one slot	7.3.4.3.15
Read Maintenance demanded in channel coding for one slot	7.3.4.3.16
Read Maintenance demanded in all codings for one slot	7.3.4.3.17
Read Diagnosis, Maintenance, Qualified and Status for one slot	7.3.4.3.18
Read Diagnosis in channel coding for one AR	7.3.4.3.19
Read Diagnosis in all codings for one AR	7.3.4.3.20
Read Maintenance required in channel coding for one AR	7.3.4.3.21
Read Maintenance required in all codings for one AR	7.3.4.3.22
Read Maintenance demanded in channel coding for one AR	7.3.4.3.23
Read Maintenance demanded in all codings for one AR	7.3.4.3.24
Read Diagnosis, Maintenance, Qualified and Status for one AR	7.3.4.3.25
Read Diagnosis in channel coding for one API	7.3.4.3.26
Read Diagnosis in all codings for one API	7.3.4.3.27
Read Maintenance required in channel coding for one API	7.3.4.3.28
Read Maintenance required in all codings for one API	7.3.4.3.29
Read Maintenance demanded in channel coding for one API	7.3.4.3.30
Read Maintenance demanded in all codings for one API	7.3.4.3.31
Read Diagnosis, Maintenance, Qualified and Status for one API	7.3.4.3.32
Read Diagnosis, Maintenance, Qualified and Status for one device	7.3.4.3.33
Read Module Diff Block	7.3.6.3.15
Read API Data	7.3.6.3.17
Read I&M0 Filter Data	7.3.6.3.18
Read I&M0 Data	7.3.6.3.19

Nom de service	Paragraphe
Read I&M1 Data	7.3.6.3.21
Read I&M2 Data	7.3.6.3.23
Read I&M3 Data	7.3.6.3.25
Read I&M4 Data	7.3.6.3.27
Read Autoconfiguration Data	7.3.6.3.28
Read Expected Identification	7.3.6.3.8
Read Real Identification	7.3.6.3.10
Read Isochronous Mode Data	7.3.7.3.2
Read PDev Data	7.3.8.3.1
Read PD Real Data	7.3.8.3.2
Read PD Expected Data	7.3.8.3.3
Read PD Interface Data Real	7.3.8.3.4
Read PD Interface Adjust	7.3.8.3.6
Read PD Interface MRP Data Real	7.3.8.3.7
Read PD Interface MRP Data Check	7.3.8.3.18
Read PD Interface MRP Data Adjust	7.3.8.3.20
Read PD Interface FSU Data Adjust	7.3.8.3.22
Read PD NC Data Check	7.3.8.3.24
Read PD Sync Data	7.3.8.3.10
Read PD IR Data	7.3.8.3.8
Read PD IR Subframe Data	7.3.8.3.13
Read PD Port Data Real	7.3.8.3.25
Read PD Port Data Check	7.3.8.3.27
Read PD Port Data Adjust	7.3.8.3.29
Read PD Port FO Data Real	7.3.8.3.30
Read PD Port FO Data Check	7.3.8.3.32
Read PD Port FO Data Adjust	7.3.8.3.34
Read PD Port MRP Data Real	7.3.8.3.35
Read PD Port MRP Data Adjust	7.3.8.3.37
Read PD Port Statistic	7.3.8.3.38
Read PD Time Data	7.3.8.3.15
Read AR Data	7.3.9.5.1

Result (-)

Ce paramètre indique que la demande de service a échoué.

Error Decode

Ce paramètre sélectionne un schéma d'erreur (Error scheme) pour Error code 1 et Error code 2; son codage est spécifié dans la CEI 61158-6-10.

Type: Unsigned8

Valeur permise: PNIORW

Error code 1

L'erreur de code (Error code) 1 prend l'une des valeurs suivantes:

read error (erreur de lecture), module failure (défaillance de module), version conflict (conflit de versions), feature not supported (caractéristique non prise en charge), user specific (spécifique à un utilisateur), invalid index (indice non valide), invalid slot/subslot (baie/sous-baie non valide), type conflict (conflit de types), invalid area (zone non valide),

state conflict (conflit d'états), access denied (accès refusé), invalid range (plage non valide), invalid parameter (paramètre non valide), invalid type (type non valide), read constrain conflict (conflit de contraintes en lecture), resource busy (ressource occupée), resource unavailable (ressource non disponible), service cancelled (service annulé).

Type: Unsigned16

Error code 2

Le paramètre Error Code 2 est spécifique à un utilisateur.

Type: Unsigned8

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 2 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 3 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

La réponse négative doit contenir les valeurs Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" si le sous-module adressé ne contient que des données de sortie.

7.3.1.4.1.2 Read Query

Ce service confirmé facultatif peut être utilisé pour lire la valeur d'un objet Record Data. Pour optimiser l'accès à plusieurs petits objets de données, un serveur peut proposer différentes combinaisons de ces données sous un seul indice Index. Dans ce cas, le contenu de l'objet Record Data est défini par le sélecteur Selector. Ce service doit être utilisé seulement conjointement à l'AR implicite (Implicit AR), IO AR ou Supervisor AR. Le Tableau 110 montre les paramètres du service.

L'attribut Read Partial Access commande le fonctionnement de la réponse lorsque la longueur demandée est plus courte que la longueur de l'objet Record Data (voir la description des attributs).

Tableau 110 – Read Query

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
API	M	M(=)		
Target AR UUID	U	U(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Index	M	M(=)		
Seq Number	M	M(=)		
Longueur	M	M(=)		
Selector	M	M(=)		
Result(+)			S	S(=)

Nom de paramètre	req	ind	rsp	cnf
AREP			M	M(=)
Seq Number			M	M(=)
Longueur			M	M(=)
Data (Données)			M	M(=)
Result (-)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

API

Si le domaine d'application de l'objet Record Data est une sous-baie, une baie ou un API, ce paramètre doit être utilisé pour adresser l'AP désiré.

Target AR UUID

Si le domaine d'application de l'objet Record Data est la relation entre applications, ce paramètre doit être utilisé conjointement à l'AR implicite (Implicit AR) pour adresser l'AR souhaitée au sein de l'appareil E/S.

Slot Number

Si le domaine d'application de l'objet Record Data est une sous-baie ou une baie, ce paramètre doit être utilisé pour adresser la baie désirée (habituellement un module).

Subslot Number

Si le domaine d'application de l'objet Record Data est une sous-baie, ce paramètre doit être utilisé pour adresser la sous-baie désirée (habituellement un sous-module).

Indice

Le paramètre Index doit être utilisé pour adresser l'objet Record Data désiré. Il est en outre utilisé pour déduire le domaine d'application d'un objet Record Data.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16}-1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de Seq Number doit être incrémenté pour chaque demande de service au cours de la session.

NOTE 1 Si, par exemple, une demande de service d'écriture a le Seq Number mis à «17», une demande de service de lecture suivante doit avoir le Seq Number mis à «18».

En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Une confirmation avec un Seq Number qui n'est pas en cours sera ignorée. Une indication avec un Seq Number qui n'est pas en cours sera rejetée. Le Seq Number doit être maintenu séparément pour chaque AR établie.

Length

Le paramètre Length indique le nombre d'octets d'un objet Record Data qui doit être lu. La longueur permise se situe dans la plage 0 à $2^{32}-256$.

Selector

Le paramètre Selector est utilisé par le serveur pour identifier les données demandées. Pour optimiser l'accès à plusieurs petits objets de données, un serveur peut proposer différentes combinaisons de ces données sous un seul indice Index et sélectionner la collection demandée avec le sélecteur Selector.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Length

Le paramètre Length (longueur) indique le nombre d'octets de l'objet Data qui a été lu.

Data

Le paramètre Data contient la valeur de l'objet qui a été lue et est constitué du nombre d'octets indiqués dans le paramètre Length. Ce paramètre doit être constitué des types de données définis à l'Article 5.

Result (-)

Ce paramètre indique que la demande de service a échoué.

Error Decode

Ce paramètre sélectionne un schéma d'erreur (Error scheme) pour Error code 1 et Error code 2; son codage est spécifié dans la CEI 61158-6-10.

Type: Unsigned8

Valeur permise: PNORW

Error code 1

L'erreur de code (Error code) 1 prend l'une des valeurs suivantes:

read error (erreur de lecture), module failure (défaillance de module), version conflict (conflit de versions), feature not supported (caractéristique non prise en charge), user specific (spécifique à un utilisateur), invalid index (indice non valide), invalid slot/subslot (baie/sous-baie non valide), type conflict (conflit de types), invalid area (zone non valide), state conflict (conflit d'états), access denied (accès refusé), invalid range (plage non valide), invalid parameter (paramètre non valide), invalid type (type non valide), read constrain conflict (conflit de contraintes en lecture), resource busy (ressource occupée), resource unavailable (ressource non disponible), service cancelled (service annulé).

Type: Unsigned16

Error code 2

Le paramètre Error Code 2 est spécifique à un utilisateur.

Type: Unsigned8

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 2 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 3 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

La réponse négative doit contenir les valeurs Error Decode = "PNORW", Error code 1 = "invalid index", Error code 2 = "user specific" si le sous-module adressé ne contient que des données de sortie.

7.3.1.4.2 Services d'écriture

7.3.1.4.2.1 Write

Ce service confirmé peut être utilisé pour écrire la valeur d'un objet Record Data. Ce service doit seulement être utilisé conjointement à l'IO AR ou à la Supervisor AR. En plus de ce service d'écriture commun, des services supplémentaires sont définis pour des données d'application spécifiques à l'écriture. Tous ces services d'écriture utilisent les mêmes paramètres, seul le champ de données est différent. Le Tableau 111 indique les paramètres du service. Ce service peut également être utilisé pour écrire les données d'un utilisateur et d'un profil supplémentaire.

L'attribut Write Partial Access commande le fonctionnement de la réponse lorsque la longueur demandée est plus courte que la longueur de l'objet Record Data (voir la description des attributs).

Tableau 111 – Write

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Index	M	M(=)		
Seq Number	M	M(=)		
Longueur	M	M(=)		
Data (Données)	M	M(=)		
Prm Flag		M		
Result(+)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Result (-)			S	S(=)
AREP			M	M(=)
Seq Number			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

API

Si le domaine d'application de l'objet Record Data est une sous-baie, une baie ou un API, ce paramètre doit être utilisé pour adresser l'AP désiré.

Slot Number

Si le domaine d'application de l'objet Record Data est une sous-baie ou une baie, ce paramètre doit être utilisé pour adresser la baie désirée (habituellement un module).

Subslot Number

Si le domaine d'application de l'objet Record Data est une sous-baie, ce paramètre doit être utilisé pour adresser la sous-baie désirée (habituellement un sous-module).

Indice

Le paramètre Index doit être utilisé pour adresser l'objet Record Data désiré. Il est en outre utilisé pour déduire le domaine d'application d'un objet Record Data.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16}-1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de Seq Number doit être incrémenté pour chaque demande de service au cours de la session.

NOTE 1 Si, par exemple, une demande de service d'écriture a le Seq Number mis à «17», une demande de service de lecture suivante doit avoir le Seq Number mis à «18».

En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Une confirmation avec un Seq Number qui n'est pas en cours sera ignorée. Une indication avec un Seq Number qui n'est pas en cours sera rejetée. Le Seq Number doit être maintenu séparément pour chaque AR établie.

Length

Le paramètre Length indique le nombre d'octets d'un objet Record Data qui doit être écrit. La longueur permise se situe dans la plage 0 à $2^{32}-256$.

Data

Le paramètre Data contient la valeur de l'objet Record Data qui doit être écrite et est constitué du nombre d'octets indiqués dans le paramètre Length de la demande. Ce paramètre doit être constitué des types de données définis à l'Article 5. Il existe une signification particulière pour les différents services de ces données, elle est décrite dans les parties concernant les services spéciaux. Le Tableau 112 indique les paramètres des services de lecture spéciaux.

Tableau 112 – Services Write (écriture)

Nom de service	Paragraphe
Write Substitute Value	7.3.2.5.18
Write I&M1 Data	7.3.6.3.20
Write I&M2 Data	7.3.6.3.22
Write I&M3 Data	7.3.6.3.24
Write I&M4 Data	7.3.6.3.26
Write Isochronous Mode Data	7.3.7.3.1
Write PD Interface Adjust	7.3.8.3.5
Write PD IR Data	7.3.8.3.7
Write PD IR Subframe Data	7.3.8.3.12
Write PD Sync Data	7.3.8.3.9
Write PD Interface MRP Data Check	7.3.8.3.17
Write PD Interface MRP Data Adjust	7.3.8.3.19
Write PD Interface FSU Data Adjust	7.3.8.3.21
Write PD NC Data Check	7.3.8.3.23
Write MRP Port Data Adjust	7.3.8.3.36
Write PD Port Data Check	7.3.8.3.26
Write PD Port Data Adjust	7.3.8.3.28

Nom de service	Paragraphe
Write PD Port FO Data Check	7.3.8.3.32
Write PD Port FO Data Adjust	7.3.8.3.33
Write PD Port MRP Data Adjust	7.3.8.3.36
Write PD Time Data	7.3.8.3.14

Prm Flag

Le paramètre d'indication local doit contenir la valeur TRUE pendant la paramétrisation au cours de la phase d'établissement de connexion. Autrement, la valeur doit être mise à FALSE.

Type d'attribut: Boolean

NOTE 2 Ce paramètre est utilisé pour commander le stockage persistant des données. Il est utilisé dans la procédure de démarrage rapide.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Decode

Ce paramètre sélectionne un schéma d'erreur (Error scheme) pour Error code 1 et Error code 2; son codage est spécifié dans la CEI 61158-6-10.

Type: Unsigned8

Valeur permise: PNORW

Error code 1

L'erreur de code (Error code) 1 prend l'une des valeurs suivantes:

read error (erreur de lecture), module failure (défaillance de module), version conflict (conflit de versions), feature not supported (caractéristique non prise en charge), user specific (spécifique à un utilisateur), invalid index (indice non valide), invalid slot/subslot (baie/sous-baie non valide), type conflict (conflit de types), invalid area (zone non valide), state conflict (conflit d'états), access denied (accès refusé), invalid range (plage non valide), invalid parameter (paramètre non valide), invalid type (type non valide), read constrain conflict (conflit de contraintes en lecture), resource busy (ressource occupée), resource unavailable (ressource non disponible), service cancelled (service annulé).

Type: Unsigned16

Error code 2

Le paramètre Error Code 2 est spécifique à un utilisateur.

Type: Unsigned8

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 3 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 4 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

La réponse négative doit contenir les valeurs Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" si le sous-module adressé ne contient que des données de sortie.

7.3.1.4.2.2 Write Combined Object Container

Ce service confirmé peut être utilisé pour écrire le conteneur d'objet combiné dans le point d'accès de l'objet combiné (voir 7.1.4.5.6). Ce service doit être utilisé conjointement à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service d'écriture (voir 7.3.1.4.2.1). Combined Object Container appartient à un sous-module, qui est adressé par l'API par défaut (zéro) et son numéro de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données du conteneur d'objet combiné tel que spécifié dans la CEI 61158-6-10.

Le paramètre Data a une signification et une structure spécifiques à un service. Le Tableau 113 indique les éléments spécifiques de Data pour ce service.

L'utilisateur du service doit utiliser ce service seulement si l'AR utilisée est connectée au sous-module adressé, qui doit être le point d'accès de l'objet combiné. Les sous-modules adressés par les attributs API, Slot Number et Subslot Number d'une entrée de conteneur particulière doivent faire partie de l'objet combiné. L'indice doit être pris en charge par le sous-module. Autrement, le fournisseur de service doit émettre une réponse négative au service et toutes les données doivent être ignorées.

Tableau 113 – Éléments Data de Write PD Time Data

N om de paramètre	éléments spécifiques à un service	req	ind
Data	List of Container	M	M(=)
	API	M	M(=)
	Slot	M	M(=)
	Subslot	M	M(=)
	Index	M	M(=)
	Data	M	M(=)

Data

Ce paramètre contient la valeur de l'objet à écrire.

List of Container

Le service envoie un certain nombre d'enregistrements de données avec cette liste à écrire dans le point d'accès. Chaque élément de liste est constitué des éléments suivant:

API

Si le domaine d'application de l'objet Record Data est une sous-baie, une baie ou un API, ce paramètre doit être utilisé pour adresser l'AP désiré.

Slot Number

Si le domaine d'application de l'objet Record Data est une sous-baie ou une baie, ce paramètre doit être utilisé pour adresser la baie désirée (habituellement un module).

Subslot Number

Si le domaine d'application de l'objet Record Data est une sous-baie, ce paramètre doit être utilisé pour adresser la sous-baie désirée (habituellement un sous-module).

Index

Le paramètre Index doit être utilisé pour adresser l'objet Record Data désiré. Il est en outre utilisé pour déduire le domaine d'application d'un objet Record Data.

Length

Le paramètre Length indique le nombre d'octets d'un objet Record Data qui doit être écrit. La longueur permise se situe dans la plage 0 à $2^{32}-256$.

Data

Le paramètre Data contient la valeur de l'objet Record Data qui doit être écrite et est constitué du nombre d'octets indiqués dans le paramètre Length de la demande. Ce paramètre doit être constitué des types de données définis à l'Article 5. Il existe une signification particulière pour les différents services de ces données, elle est décrite dans les parties concernant les services spéciaux. Le Tableau 112 indique les paramètres des services de lecture spéciaux.

7.3.1.4.2.3 Local Write Multiple

Ce service local peut être utilisé pour acheminer plusieurs objets Record Data au sein d'une seule et même APDU. Ce service doit seulement être utilisé conjointement à l'IO AR ou à la Supervisor AR. Le Tableau 114 montre les paramètres du service.

NOTE 1 Il est possible d'acheminer des objets Record Data et les attributs issus de plusieurs ASE (par exemple l'ASE Physical Device Management) au sein d'une seule APDU.

Ce service local est mappé en construisant un enregistrement de données voisines à partir des informations de chaque adresse (API, baie, à sous-baie, indice), du numéro de séquence et des données des enregistrements de données particuliers. Cet enregistrement de données est ensuite transmis à l'appareil grâce au service Write.req (voir 7.3.1.4.2.1).

Dans un appareil, cet enregistrement de données voisines est reconnu par son indice spécial (voir la CEI 61158-6-10 pour sa définition, en particulier la substitution APDU IODWrite_Multiple_Req). Dans l'appareil, la réception d'un tel enregistrement de données (indiqué par Write.ind) est ensuite mappée au service local Local New Write Multiple.

Tableau 114 – Local Write Multiple

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
Seq Number	M	
List of Data Records	M	
API	M	
Slot Number	M	
Subslot Number	M	
Index	M	
Seq Number	M	
Longueur	M	
Data (Données)	M	
Prm Flag		
Result(+)		S
AREP		M
Seq Number		M
List of Results		M
Seq Number		M
PNIO Status		M
Add Data 1		M
Add Data 2		M

Nom de paramètre	req	cnf
Result (-)		S
AREP		M
Seq Number		M
Error Decode		M
Error code 1		M
Error code 2		M
Add Data 1		M
Add Data 2		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16} - 1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de valeur Seq Number doit être incrémenté pour chaque demande de service au cours de la session. En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Les confirmations avec un Seq Number qui n'est pas en cours seront ignorées. Les indications avec un Seq Number qui n'est pas en cours seront rejetées. Le Seq Number doit être maintenu séparément pour chaque AR établie.

NOTE 2 Ce numéro de séquence est utilisé pour le service «write multiple».

List of Data Records

Le service envoie un certain nombre d'enregistrements de données avec cette liste. Chaque élément de liste est constitué des éléments suivants:

API

Ce paramètre doit être utilisé pour adresser l'API souhaité.

Slot Number

Le paramètre Slot Number est utilisé dans l'appareil de destination pour adresser l'objet Record Data souhaité dans une baie spécifique (typiquement un module).

Subslot Number

Le paramètre Subslot Number est utilisé dans l'appareil de destination pour adresser l'objet Record Data souhaité dans un sous-élément spécifique (typiquement un sous-module).

Indice

Le paramètre Index est utilisé dans l'appareil de destination pour adresser l'objet Record Data voulu.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16} - 1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de valeur Seq Number doit être incrémenté pour chaque demande de service au cours de la session. En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Les confirmations avec un Seq Number qui n'est pas en cours seront ignorées. Les indications avec un Seq Number qui n'est pas en cours seront rejetées. Le Seq Number doit être maintenu séparément pour chaque AR établie.

NOTE 3 Ce numéro de séquence est utilisé pour chaque enregistrement de données.

Length

Le paramètre Length indique le nombre d'octets d'un objet Record Data qui doit être écrit. La longueur permise se situe dans la plage 0 à $2^{32}-65$.

Data

Le paramètre Data contient la valeur de l'objet Record Data qui doit être écrite et est constitué du nombre d'octets indiqués dans le paramètre Length de la demande. Ce paramètre doit être constitué des types de données définis à l'Article 5.

Prm Flag

Le paramètre d'indication local doit contenir la valeur TRUE pendant la paramétrisation au cours de la phase d'établissement de connexion. Autrement, la valeur doit être mise à FALSE.

Type d'attribut: Boolean

NOTE 4 Ce paramètre est utilisé pour commander le stockage persistant des données. Il est utilisé dans la procédure de démarrage rapide.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.1.4.2.4 Local New Write Multiple

Ce service local indique la réception de plusieurs objets Record Data au sein d'une seule APDU. Ce service doit seulement être utilisé conjointement à l'IO AR ou à la Supervisor AR. Le Tableau 115 montre les paramètres du service.

NOTE 1 Il est possible d'acheminer des objets Record Data et les attributs issus de plusieurs ASE (par exemple l'ASE Physical Device Management) au sein d'une seule APDU.

Ce service local est mappé par la réception d'un enregistrement de données de voisinage indiqué par la réception de l'indice spécial (voir la CEI 61158-6-10 pour sa définition; en particulier la substitution d'APDU IODWrite_Multiple_Req) sous l'indication de service Write.ind (voir 7.3.1.4.2.1).

Les informations de chaque adresse (API, baie, à sous-baie, indice), le numéro de séquence et des données des enregistrements de données particuliers sont extraits de l'enregistrement de données de voisinage par l'application et présentés dans l'indication de ce service.

Si l'enregistrement de données de voisinage n'est pas décodable (en raison de sa structure interne), un result(-) est renvoyé.

Sinon, les enregistrements de données particulières sont appliqués un à un sur l'application en utilisant les informations d'adresse et les données. Les résultats des écritures particulières d'enregistrements de données sont rassemblées (voir la CEI 61158-6-10 pour sa définition; en particulier la substitution d'APDU IODWrite_Multiple_Req) et retournées par l'intermédiaire de Write.rsp(+).

Le service local Write Multiple.cnf extrait ensuite les résultats particuliers de l'information renvoyée et il les renvoie à l'application.

Tableau 115 – Local New Write Multiple

Nom de paramètre	ind	rsp
Argument	M(=)	
AREP	M(=)	
Seq Number	M(=)	
Slot Number	M(=)	
Subslot Number	M(=)	
Index	M(=)	
Seq Number	M(=)	
Longueur	M(=)	
Data (Données)	M(=)	
Prm Flag	M	
Result(+)		S
AREP		M
Seq Number		M
List of Results		M
Seq Number		M
PNIO Status		M
Add Data 1		M
Add Data 2		M
Result(-)		S
AREP		M
Seq Number		M
Error Decode		M
Error code 1		M
Error code 2		M
Add Data 1		M
Add Data 2		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16}-1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de valeur Seq Number doit être incrémenté pour chaque demande de service au cours de la session. En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Les confirmations avec un Seq Number qui n'est pas en cours seront ignorées. Les indications avec un Seq Number qui n'est pas en cours seront rejetées. Le Seq Number doit être maintenu séparément pour chaque AR établie.

NOTE 2 Ce numéro de séquence est utilisé pour le service «write multiple».

List of Data Records

Le service envoie un certain nombre d'enregistrements de données avec cette liste. Chaque élément de liste est constitué des éléments suivants:

API

Ce paramètre doit être utilisé pour adresser l'API souhaité.

Slot Number

Le paramètre Slot Number est utilisé dans l'appareil de destination pour adresser l'objet Record Data souhaité dans une baie spécifique (typiquement un module).

Subslot Number

Le paramètre Subslot Number est utilisé dans l'appareil de destination pour adresser l'objet Record Data souhaité dans un sous-élément spécifique (typiquement un sous-module).

Indice

Le paramètre Index est utilisé dans l'appareil de destination pour adresser l'objet Record Data voulu.

Seq Number

Le paramètre Seq Number est utilisé par le serveur pour identifier la duplication de services avec un numéro de séquence. La plage de ce paramètre de service va de 0 à $2^{16} - 1$. Le processus d'application demandeur doit fournir un unique Seq Number à chaque demande de services en cours. Le paramètre de valeur Seq Number doit être incrémenté pour chaque demande de service au cours de la session. En commençant une nouvelle session, le numéro de séquence doit commencer par la dernière valeur de la session précédente. Les confirmations avec un Seq Number qui n'est pas en cours seront ignorées. Les indications avec un Seq Number qui n'est pas en cours seront rejetées. Le Seq Number doit être maintenu séparément pour chaque AR établie.

NOTE 3 Ce numéro de séquence est utilisé pour chaque enregistrement de données.

Length

Le paramètre Length indique le nombre d'octets d'un objet Record Data qui doit être écrit. La longueur permise se situe dans la plage 0 à $2^{32} - 65$.

Data

Le paramètre Data contient la valeur de l'objet Record Data qui doit être écrite et est constitué du nombre d'octets indiqués dans le paramètre Length de la demande. Ce paramètre doit être constitué des types de données définis à l'Article 5.

Prm Flag

Le paramètre d'indication local doit contenir la valeur TRUE pendant la paramétrisation au cours de la phase d'établissement de connexion. Autrement, la valeur doit être mise à FALSE.

Type d'attribut: Boolean

NOTE 4 Ce paramètre est utilisé pour commander le stockage persistant des données. Il est utilisé dans la procédure de démarrage rapide.

Result (+)

Ce paramètre indique que la demande de service elle-même a réussi; noter cependant que le statut PNIO (PNIO Status) d'une partie ou même de tous les résultats d'écriture peut indiquer une erreur.

List of Results

Cette liste contient le résultat de chaque accès aux enregistrements de données. Chaque élément de liste est constitué des éléments suivants:

PNIO Status

Ce paramètre indique le type d'erreur; voir 7.3.1.3.5 pour les détails.

Type: Unsigned32

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 5 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 6 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

Result(-)

Ce paramètre indique que la demande de service a totalement échoué et qu'aucun des enregistrements de données n'a été écrit.

PNIO Status

Ce paramètre indique le type d'erreur; 7.3.1.3.5 pour les détails.

Type: Unsigned32

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 7 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 8 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

7.3.2 ASE IO Data

7.3.2.1 Aperçu général

Dans l'environnement de couche application, chaque processus d'application d'un appareil E/S doit contenir exactement un seul objet IO Data pour chaque instance de processus d'application. Il est structuré au moyen de baies et de sous-baies. Les sous-baies représentent des données E/S qui doivent être acheminées de manière cyclique sur le réseau. Par conséquent, l'ASE IO Data définit les attributs des objets IO Data et fournit un jeu de services utilisés pour récupérer et établir le tampon pour transporter leurs valeurs. Les objets IO Data (données E/S) sont séparés en objet Input Data (Données d'entrée) et objet Output Data (Données de sortie).

Un objet Input Data ou un objet Output Data doit être partitionné en Input Data Elements (éléments de données d'entrée) ou Output Data Elements (éléments de données de sortie) qui représentent les IO Data des baies/sous-baies.

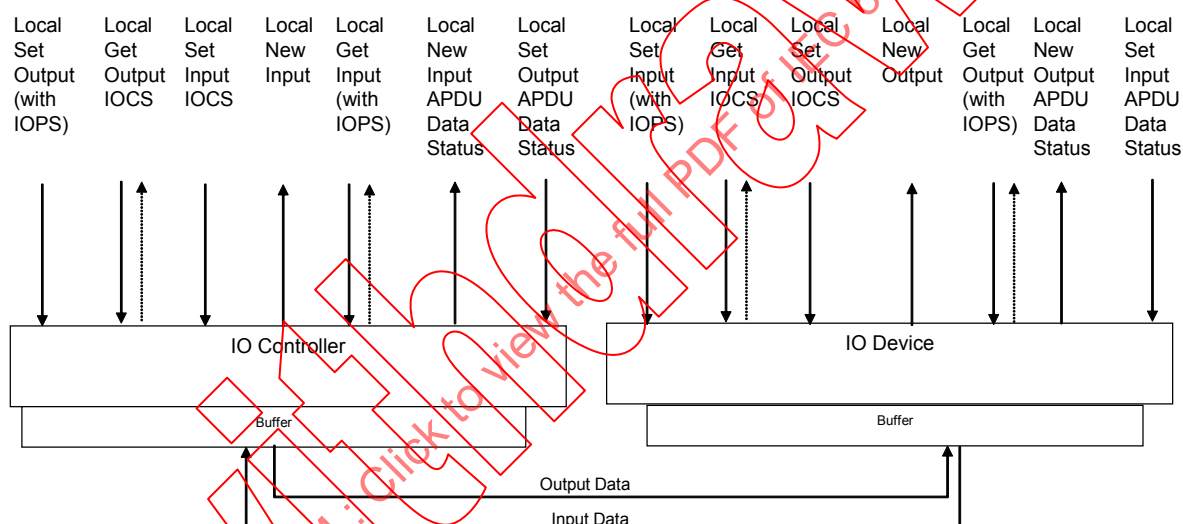
Des services supplémentaires sont fournis pour lire de façon acyclique les valeurs des objets IO Data et pour indiquer les nouvelles valeurs pour l'objet Input Data et l'objet Output Data. De plus, des services sont fournis pour indiquer la présence de nouveaux objets Input Data consommés et en récupérer le contenu.

Les objets IO Data sont adressés de façon implicite par l'intermédiaire des services connexes. La granularité des données d'entrée ou de sortie dans un serveur/fournisseur est fonction des attributs de configuration correspondants.

L'ASE I/O Data utilise le modèle d'accès Client/Serveur et Fournisseur/Consommateur. Le modèle Client/Serveur est caractérisé en ce qu'une application Client achemine la valeur pour l'objet Output Data vers le tampon du Serveur. L'application Serveur récupère cette valeur par un service Get Output. La réception d'une nouvelle valeur est indiquée par le service New Output. L'application Serveur achemine la valeur de son objet Input Data par le service Set Input vers le tampon du Serveur dès que l'une des valeurs d'objet réel change. L'application Client récupère cette valeur par le service Get Input. L'émission du tampon Serveur vers le tampon Client et vice versa est effectuée de façon asynchrone par rapport aux services Get Output et Set Input.

Le modèle formel de l'ASE "IO Data" (données E/S) fait l'objet d'une description ci-après, suivie d'une description de ses services. En outre, l'ASE IO Data représente la structure réelle d'entrée et de sortie d'un appareil E/S. Les relations entre structure réelle de données E/S et acheminement de données E/S par des relations d'applications (AR) sont décrites au sein de l'ASE Context.

La Figure 26 montre les interactions de services E/S pour l'appareil de commande E/S et l'appareil E/S.



Légende

Anglais	Français
Local Set Output (with IOPS)	Etablir la sortie (avec IOPS) en local
Local Get Output IOCS	Récupérer l'IOCS de la sortie en local
Local Set Input IOCS	Etablir l'IOCS de l'entrée en local
Local New Input	Nouvelle entrée en local
Local Get Input (with IOPS)	Récupérer l'entrée (avec IOPS) en local
Local New Input APDU Data Status	Nouveau statut de données d'APDU d'entrée en local
Local Set Output APDU Data Status	Etablir le statut de données d'APDU de sortie en local
Local Set Input (with IOPS)	Etablir l'entrée (avec IOPS) en local
Local Get Input IOCS	Récupérer l'entrée IOCS en local
Local Set Output IOCS	Etablir l'IOCS de la sortie en local
Local New Output	Nouvelle sortie en local
Local Get Output (with IOPS)	Récupérer la sortie (avec IOPS) en local
Local New Output APDU Data Status	Nouveau statut de données d'APDU de sortie en local
Local Set Input APDU Data Status	Etablir le statut de données d'APDU d'entrée en local
IO Controller	Appareil de commande E/S

Anglais	Français
IO Device	Appareil E/S
Buffer	Tampon
Output Data	Données de sortie
Input Data	Données d'entrée

Figure 26 – Vue d'ensemble des interactions de services des ASE IO

Les services interagissent avec des tampons locaux maintenus par le protocole de couche application. Ce protocole fournit les mécanismes pour l'échange cyclique de tampons entre les appareils. Les services locaux fournissent un moyen de récupérer ou d'établir les valeurs des données E/S conjointement à leurs valeurs de statut. Les valeurs de statut sont appelées "IOPS" et "IOCS".

7.3.2.2 Comportement des statuts IOPS/IOCS

7.3.2.2.1 IOPS

L'IOPS (Input Output Provider Status, c'est-à-dire Statut de fournisseur d'entrées et de sorties) décrit la validité de la source de données vue par le souscripteur des données. Il appartient à Output Data pour un appareil de commande E/S et à Input Data pour un appareil E/S.

NOTE Si un sous-module ne contient pas de données d'entrée ou de sortie, il est considéré comme étant un élément Input Data avec la longueur de l'attribut Input Data mise à zéro.

L'instance détectrice doit positionner l'IOPS en conséquence et immédiatement. L'IOPS est défini par des moyens locaux; par exemple: si erreurs de configuration, si le sous-module ou module «tiré», si état de paramétrage insuffisant du sous-module (application prête en attente), si diagnostic.

L'IOPS peut être défini par les instances individuelles participant à la transmission des données d'entrée: par le sous-module (dont le capteur), par le module, par l'appareil E/S et l'appareil de commande E/S (localement, par exemple: temporisation de maintien de données).

L'IOPS permet d'identifier la première instance qui a détecté le contenu non valide, avec l'aide des valeurs suivantes:

- GOOD – les données sont valides
- BAD_BY_SUBSLOT – les données ne sont pas valides telles qu'elles sont détectées par le sous-module
- BAD_BY_SLOT – les données ne sont pas valides telles qu'elles sont détectées par le module
- BAD_BY_DEVICE – les données ne sont pas valides telles qu'elles sont détectées par l'appareil E/S
- BAD_BY_CONTROLLER – les données ne sont pas valides telles qu'elles sont détectées par l'appareil de commande E/S

Seules des données valides avec IOPS==GOOD doivent être traitées par l'application. Dans tous les autres cas (BAD_BY_XXX), l'application doit générer une erreur en local (ex: sur le programme d'application de l'appareil E/S) ou utiliser des valeurs par défaut (par exemple: zéro, dernière valeur valide, valeur substitut sur l'appareil E/S) selon sa configuration et sa paramétrisation..

L'IOPS doit être établi indépendamment de l'IOCS correspondant; par exemple: si l'IOCS est mis sur BAD_BY_CONTROLLER, mais que l'appareil E/S est toujours capable d'envoyer des données valides, l'IOPS doit être mis sur GOOD.

7.3.2.2.2 IOCS

L'IOCS (Input Output Consumer Status, c'est-à-dire Statut du consommateur d'objet d'entrée/sortie) indique si l'application (appareil de commande ou appareil) traite la valeur. Il appartient à Input Data pour un appareil de commande E/S et à Output Data pour un appareil E/S.

L'IOCS est défini par les instances individuelles participant à la transmission des données de sortie: par le sous-module (dont l'acteur), par le module, par l'appareil E/S et l'appareil de commande E/S (pour un module d'entrée en local, ex: temporisation de maintien de données; pour un module de sortie ex: du fait qu'il soit à l'état de fonctionnement Stop).

L'instance détectrice doit positionner l'IOCS en conséquence et immédiatement. L'IOCS est défini par des moyens locaux; par exemple: si erreurs de configuration, si le sous-module ou module «tiré», si état de paramétrage insuffisant du sous-module (application prête en attente), si diagnostic.

L'IOCS permet d'identifier la première instance qui ne traite plus le contenu, avec l'aide des valeurs suivantes:

- GOOD – les données sont traitées par l'application
- BAD_BY_SUBSLOT – les données ne sont pas traitées par l'application telles qu'elles sont détectées par le sous-module
- BAD_BY_SLOT – BAD_BY_SLOT – les données ne sont pas traitées par l'application telles qu'elles sont détectées par le module
- BAD_BY_DEVICE – les données ne sont pas traitées par l'application telles qu'elles sont détectées par l'appareil
- BAD_BY_CONTROLLER – les données ne sont pas traitées par l'application telles qu'elles sont détectées par le sous-module.

Pour un appareil E/S ce qui suit s'applique: L'IOCS doit être établi indépendamment de l'IOPS correspondant; autrement dit, si l'IOPS est mis sur BAD_BY_xxx mais que l'appareil E/S est toujours en mesure de traiter des données, l'IOCS doit être laissé sur GOOD.

Pour un appareil de commande E/S, ce qui suit s'applique: L'IOCS peut être établi dépendamment ou indépendamment de l'IOPS correspondant. L'action possible de l'appareil de commande E/S ne relève pas du domaine d'application de cette définition de services.

NOTE Le modèle des meilleures pratiques pour la gestion au sein de l'appareil de commande E/S est le suivant: Si l'appareil E/S ne peut pas traiter les données de sortie (indiqué en mettant IOCS sur «BAD_BY_XXX»), il convient que l'application de l'appareil de commande E/S mette l'IOPS correspondant à «BAD_BY_CONTROLLER». Cela permet à la demande de l'appareil de commande E/S d'appliquer les valeurs initiales lorsque l'appareil E/S peut à nouveau traiter les données de sortie; indiqué en mettant l'IOCS sur «GOOD» et en envoyant une alarme «return of submodule». Au cours du processus de l'alarme au sein de l'application de l'appareil de commande E/S, la valeur initiale est appliquée aux données de sortie et, par conséquent, l'IOPS est mis sur «GOOD» à nouveau.

L'action possible de l'appareil E/S ne relève pas du domaine d'application de cette définition de services.

7.3.2.2.3 Comportement lors des services Connect, Plug, Prm Begin et Application Ready

Si AR Info a répondu dans le Module Diff Block pour un sous-module particulier à l'état APPLICATION_READY_PENDING, l'IOPS/IOCS correspondant doit être établi sur BAD_BY_SUBSLOT.

L'IOPS/IOCS doit être établi sur GOOD au plus tôt après le paramétrage du sous-module et à la condition qu'aucune autre circonstance (ex: application) n'entrave la validité des données.

Si AR Info a répondu dans le Module Diff Block pour un sous-module particulier à l'état SUPERORDINATED_LOCKED, LOCKED_BY_IO_CONTROLLER ou LOCKED_BY_IO_SUPERVISOR, l'IOPS/IOCS correspondant doit être établi sur BAD_BY_SUBSLOT.

L'IOPS/IOCS doit être établi sur GOOD au plus tôt si la relation d'application obtient la propriété et à la condition qu'aucune autre circonstance (ex: application) n'entrave la validité des données.

Si Ident Info a répondu dans le Module Diff Block pour un sous-module particulier à l'état WRONG ou NO, l'IOPS/IOCS correspondant doit être établi sur BAD_BY_SUBSLOT.

L'IOPS/IOCS doit être établi sur GOOD au plus tôt après le branchement du sous-module approprié, si la relation d'application obtient la propriété et à la condition qu'aucune autre circonstance (ex: application) n'entrave la validité des données.

Si Module State a répondu dans le Module Diff Block pour un sous-module particulier à l'état WRONG_MODULE ou NO_MODULE, tous les IOPS/IOCS correspondants des sous-modules de ce module doivent être établis sur BAD_BY_SLOT.

L'IOPS/IOCS doit être établi sur GOOD au plus tôt pour un sous-module particulier après le branchement du sous-module approprié à ce sous-module, si la relation d'application obtient la propriété et à la condition qu'aucune autre circonstance (ex: application) n'entrave la validité des données.

7.3.2.2.4 Comportement lors des transitions

Si un appareil E/S modifie un IOPS/IOCS particulier, il doit indiquer une raison à l'appareil de commande E/S.

- Lors de la transition de GOOD à BAD_BY_XXX:
 - Si la propriété du sous-module a été transférée à un superviseur E/S ou à l'application (ex: superordonnée et verrouillée), la raison est indiquée par une alarme Controlled By Supervisor.
 - Si le sous-module a été extrait, la raison est indiquée par une alarme Pull.
 - Si l'application indique la non validité des données E/S, la raison est indiquée par exemple par une alarme de diagnostic en mettant ProviderState sur « Stop »...
- Lors de la transition de BAD_BY_XXX à GOOD:
 - Si la propriété du sous-module a été restituée à l'appareil de commande E/S, la raison est indiquée par une alarme Released.
 - Si le sous-module a été rebranché, la raison est indiquée par une alarme Plug.
- Si l'application indique la validité des données E/S, la raison est indiquée par une alarme de diagnostic en mettant ProviderState sur « Run »... En outre, une alarme Return of Submodule doit être indiquée.

7.3.2.3 Comportement du statut APDU

Des services sont fournis pour établir et indiquer un changement du statut de données d'application (APDU Data Status).

7.3.2.4 Spécification de la classe IO Data

7.3.2.4.1 Spécification de la classe Input Data

7.3.2.4.1.1 Modèle

L'objet Input Data est décrit par le modèle suivant:

ASE: ASE IO Data
CLASS: Input Data
CLASS ID: pas utilisé
PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Attribut-clé:	Implicit
2	(m) Attribut:	List of APs
2.1	(m) Attribut:	API
2.2	(m) Attribut:	List of Input Data Elements
2.2.1	(m) Attribut:	Slot Number
2.2.2	(m) Attribut:	List of Subslots
2.2.2.1	(m) Attribut:	Subslot Number
2.2.2.2	(m) Attribut:	Input Data Object Format
2.2.2.2.1	(m) Attribut:	Length Input Data
2.2.2.2.2	(m) Attribut:	Input Data Structure
2.2.2.3	(m) Attribut:	Length IOPS
2.2.2.4	(m) Attribut:	IOPS
2.2.2.5	(m) Attribut:	Length IOCS
2.2.2.6	(m) Attribut:	IOCS

SERVICES:

1	(m) OpsService:	Local Set Input
2	(m) OpsService:	Local Set Input IOCS
3	(m) OpsService:	Local Get Input
4	(m) OpsService:	Local Get Input IOCS
5	(o) OpsService:	Local New Input
6	(m) OpsService:	Local Set Redundancy
7	(m) OpsService:	Local Set State
8	(m) OpsService:	Local Data State Changed
9	(m) OpsService:	Read Record Input Data Object Element

7.3.2.4.1.2 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet Input Data est adressé de façon implicite par les services.

List of APs

Un AP se compose des éléments de liste suivants:

API

Cet attribut définit le processus d'application auquel appartiennent les Input Data Elements.

Type d'attribut: Unsigned32

Valeurs permises: 0 comme processus d'application par défaut. Les valeurs 1 à 0xFFFFFFFF sont réservées pour usage futur dans les lignes directrices des profils.

List of Input Data Elements

Un élément Input Data Element se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient l'Input Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient l'Input Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

NOTE Si ARProperties.PullModuleAlarmAllowed n'est pas utilisé, la sous-baie numéro zéro est utilisée conjointement à AlarmType «Pull» pour adresser un certain module dans l'appareil E/S. Dans ce cas, elle ne doit pas contenir de voies et de données E/S.

Input Data Object Format

Cet attribut est constitué des attributs suivants.

Length Input Data

L'attribut Length Input Data définit le nombre d'octets dans l'Input Data Element sans IOPS.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 1 439

Input Data Structure

Cet attribut contient soit une Simple Data Description, soit une Array Data Description, soit une Record Data Description. Pour la description de données, le modèle utilisé est celui donné dans la CEI 61158-1.

Les données d'entrée doivent être émises totalement, en étant cohérentes, du tampon d'envoi vers le tampon de réception. Cependant, l'application peut récupérer ou établir des parties seulement d'un seul objet de données incluant l'IOPS.

Ce champ fait partie du GSD.

De plus, si les Input Data sont la source d'une Multicast Provider CR (relation de communication de fournisseur en multidiffusion), les Output Data associées de la Multicast Consumer CR doivent avoir exactement la même structure de données.

Length IOPS

L'attribut Length IOPS définit le nombre d'octets du statut de fournisseur.

Type d'attribut: Unsigned8

Valeurs permises: 1, (les valeurs 2 à 255 sont réservées pour usage futur)

IOPS

Cet attribut définit le statut du fournisseur pour l'Input Data Object Element qui appartient à la sous-baie de l'appareil E/S.

Type d'attribut: Unsigned8

Valeurs permises:

GOOD – contenu d'Input Data Object Element valide,

BAD_BY_SUBSLOT – contenu d'Input Data Object Element non valide; il a été détecté par le sous-module,

BAD_BY_SLOT – contenu d'Input Data Object Element non valide; il a été détecté par le module,

BAD_BY_DEVICE – contenu d'Input Data Object Element non valide; il a été détecté par l'appareil E/S,

Cas spécial: BAD_BY_CONTROLLER - le contenu de l'Input Data Object Element n'est pas valide, seulement détecté localement par l'appareil de commande E/S lui-même, par exemple par le biais d'une temporisation de communication. Cette valeur doit seulement être utilisée avec le paramètre de service IOPS conjointement au service Get Input au niveau de l'appareil de commande E/S. Elle ne doit jamais être appliquée avec le service Set Input au niveau de l'appareil E/S.

Length IOCS

L'attribut Length IOCS définit le nombre d'octets du statut de consommateur.

Type d'attribut: Unsigned8

Valeurs permises: 1, (les valeurs 2 à 255 sont réservées pour usage futur)

IOCS

L'attribut définit le statut de consommateur pour l'élément d'objet Input Data comme une rétroaction d'application de l'appareil de commande E/S.

Type d'attribut: Unsigned8

Valeurs permises:

GOOD – l'Input Data Object Element a pu être traité avec succès par le processus d'application de l'appareil de commande E/S,

BAD_BY_CONTROLLER – l'Input Data Object Element n'a pas pu être traité avec succès par le processus d'application de l'appareil de commande E/S, par exemple dans le cas d'une erreur ou en raison de l'état de fonctionnement («stop») l'appareil de commande E/S

Cas spécial: BAD_BY_DEVICE – l'appareil E/S a détecté localement des problèmes pour acheminer des données. Il ne doit jamais être appliqué avec le service Set Input IOCS au niveau de l'appareil de commande E/S.

7.3.2.4.1.3 Invocation de l'objet Input Data

Pour l'invocation de l'objet Input Data, les règles suivantes s'appliquent:

- Les données d'éléments d'objet Input Data ne doivent pas dépasser la longueur totale de 1 439 octets. Les attributs List of Input Data Elements, Input Data Object Format respectivement Format et Length doivent être positionnés en conséquence.
- La relation d'un élément d'objet Input Data à une Input CR, Output CR (seulement IOCS), M CR sera traitée dans l'ASE Context.
- La somme de tous les éléments d'objet Input Data reliés à une Input CR ou M CR, y compris l'IOPS et l'IOCS des éléments d'objet Output Data connexes, ne doit pas dépasser la longueur totale de 1 440 octets.
- Un objet Input Data doit être invoqué dans un processus d'application d'un appareil E/S.

7.3.2.4.2 Spécification de la classe Output Data

7.3.2.4.2.1 Modèle

L'objet Output Data est décrit par le modèle suivant:

ASE: ASE IO Data

CLASS: Output Data

CLASS ID: pas utilisé

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Attribut-clé:	Implicit
2	(m) Attribut:	List of APs
2.1	(m) Attribut:	API
2.2	(m) Attribut:	List of Output Data Elements
2.2.1	(m) Attribut:	Slot Number
2.2.2	(m) Attribut:	List of Subslots
2.2.2.1	(m) Attribut:	Subslot Number
2.2.2.2	(m) Attribut:	Output Data Object Format
2.2.2.2.1	(m) Attribut:	Length Output Data
2.2.2.2.2	(m) Attribut:	Output Data Structure
2.2.2.3	(m) Attribut:	Length IOPS
2.2.2.4	(m) Attribut:	IOPS
2.2.2.5	(m) Attribut:	Length IOCS
2.2.2.6	(m) Attribut:	IOCS
2.2.2.7	(m) Attribut:	Safe State Behavior Changeable
2.2.2.8	(m) Attribut:	Substitute Mode

2.2.2.9	(m) Attribut:	Output Substitute Data Valid
2.2.2.10	(m) Attribut:	Output Substitute Data

SERVICES:

1	(m) OpsService:	Local Set Output
2	(m) OpsService:	Local Set Output IOCS
3	(m) OpsService:	Local Get Output
4	(m) OpsService:	Local Get Output IOCS
5	(o) OpsService:	Local New Output
6	(m) OpsService:	Local Set Provider State
7	(m) OpsService:	Read Record Output Data Object Element
8	(m) OpsService:	Read Substitute Value
9	(m) OpsService:	Write Substitute Value

7.3.2.4.2.2 Attributs**Implicit**

L'attribut clé « Implicit » indique que l'objet Output Data est adressé de façon implicite par les services.

List of APs

Un AP se compose des éléments de liste suivants:

API

Cet attribut définit le processus d'application auquel appartiennent les Output Data Elements.

Type d'attribut: Unsigned32

Valeurs permises: 0 comme processus d'application par défaut; 1 à 0xFFFFFFFF sont réservés pour usage dans les lignes directrices des profils.

List of Output Data Elements

Un élément Output Data Element se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient l'Output Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient l'Output Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

NOTE 1 La valeur de Subslot Number 0 est utilisée conjointement à AlarmType «Pull» pour adresser un certain module dans l'appareil E/S. Il ne s'agit jamais d'un sous-module "réel" et il ne contient pas d'Output Data Elements.

Output Data Object Format

Cet attribut est constitué des attributs suivants.

Length Output Data

L'attribut Length Output Data définit le nombre d'octets dans l'Output Data Element sans IOPS.

Type d'attribut: Unsigned16

Valeurs permises: 1 à 1 439

Output Data Structure

Cet attribut peut contenir soit une Simple Data Description, soit une Array Data Description, soit une Record Data Description. Pour la description de données, le modèle utilisé est celui donné dans la CEI 61158-1.

Les données de sortie doivent être totalement émises, en étant cohérentes, du tampon d'envoi vers le tampon de réception. Cependant, l'application peut récupérer ou établir des parties seulement d'un seul objet de données incluant l'IOPS.

Ce champ fait partie du GSD.

De plus, si les Output Data sont le puits d'une Multicast Consumer CR (relation de communication de consommateur en multidiffusion), les Input Data associées de la Multicast Consumer CR doivent avoir exactement la même structure de données

Length IOPS

L'attribut Length IOPS définit le nombre d'octets du statut de fournisseur.

Type d'attribut: Unsigned8

Valeurs permises: 1, (les valeurs 2 à 255 sont réservées pour usage futur)

IOPS

Cet attribut définit le statut du fournisseur pour l'Output Data Object Element qui appartient à la sous-baie de l'appareil E/S.

Type d'attribut: Unsigned8

Valeurs permises:

GOOD – le Output Data Object Element a été traité avec succès par le processus d'application de l'appareil de commande E/S,
BAD_BY_CONTROLLER – le Output Data Object Element n'a pas pu être traité avec succès par le processus d'application de l'appareil de commande E/S, par exemple dans le cas d'une erreur ou en raison de l'état de fonctionnement («stop») l'appareil de commande E/S

Cas spécial: BAD_BY_DEVICE – l'appareil E/S a détecté localement des problèmes pour acheminer des données. Elle ne doit jamais être appliquée avec le service Set Input au niveau de l'appareil de commande E/S.

Length IOCS

L'attribut Length IOCS définit le nombre d'octets du statut de consommateur.

Type d'attribut: Unsigned8

Valeurs permises: 1, (les valeurs 2 à 255 sont réservées pour usage futur)

IOCS

Cet attribut définit le statut du consommateur pour l'Output Data Object Element qui appartient à la sous-baie de l'appareil E/S.

Type d'attribut: Unsigned8

Valeurs permises:

GOOD – le Output Data Object Element a été traité avec succès,
BAD_BY_SUBSLOT – le Output Data Object Element n'a pas été traité avec succès, il a été détecté par le sous-module,
BAD_BY_SLOT – le Output Data Object Element n'a pas été traité avec succès, il a été détecté par le module,
BAD_BY_DEVICE – le Output Data Object Element n'a pas été traité avec succès, il a été détecté par l'appareil E/S,

Cas spécial: BAD_BY_CONTROLLER - détecté seulement localement par l'appareil de commande E/S lui-même, par exemple par le biais d'une temporisation de communication. Cette valeur doit seulement être utilisée avec le paramètre de service IOCS conjointement au service Get Output IOCS au niveau de l'appareil de commande E/S. Elle ne doit jamais être appliquée avec le service Set Output IOCS au niveau de l'appareil E/S.

Safe State Behavior Changeable

L'attribut Safe State Behavior Changeable définit si ce comportement est réajustable.

Type d'attribut: Boolean

Substitution Mode

L'attribut Substitution Mode définit le traitement de la sous-baie de données de sortie en cas de défaillance. Ce comportement doit être appliqué si la valeur d'IOPS contient d'autres valeurs que GOOD.

Type d'attribut: Unsigned16

Valeurs permises:

ZERO – est une fonction spécifique à un sous-module pour mettre toutes les valeurs de sortie de l'élément Output Data à zéro (IOPS==GOOD) ou "inactive" (IOPS==BAD),

LAST_VALUE – toutes les valeurs de sortie de l'élément Output Data doivent être mises aux dernières valeurs valides (IOPS==GOOD) values de la primitive de service "confirmation" de Get Output,

REPLACEMENT_VALUE – toutes les valeurs de sortie de l'élément Output Data doivent être mises à la valeur de l'attribut Safe State Value

PROFILE_SPECIFIC1 - PROFILE_SPECIFIC255 réservé pour spécification de profils

Substitute Data Valid

L'attribut Substitute Data Valid définit la validité de l'attribut Substitute Data.

Type d'attribut: Boolean

Valeurs

permises:

TRUE doit être utilisé conjointement à Substitution Mode ZERO, LAST_VALUE, REPLACEMENT_VALUE et PROFILE_SPECIFIC pour indiquer des données valides.

FALSE doit être utilisé conjointement à Substitution Mode ZERO et PROFILE_SPECIFIC pour indiquer des données non valides. Dans ce cas, le contenu de données doit être mis à zéro.

Substitute Data

L'attribut Substitute Data définit la valeur de remplacement des données de sortie pour la sous-baie dans le cas d'une défaillance.

Le type d'attribut doit être le même que celui défini dans l'attribut Output Data Structure.

La valeur de l'attribut Substitute Data doit être la dernière valeur valide de l'élément Output Data dans le cas de Substitution Mode LAST_VALUE.

7.3.2.4.2.3 Invocation de l'objet Output Data

Pour l'invocation de l'objet Output Data, les règles suivantes s'appliquent:

- Les données d'éléments d'objet Output Data (sous-baie) ne doivent pas dépasser la longueur totale de 1 439 octets. Les attributs List of Output Data Elements, Output Data Object Format respectivement Format et Length doivent être positionnés en conséquence.
- La relation d'un élément d'objet Output Data à une Output CR, Input CR, M CR sera traitée dans l'ASE Context.
- La somme de tous les éléments d'objet Output Data, y compris l'IOPS et l'IOCS des éléments d'objet Input Data connexes reliés à une CR, ne doit pas dépasser la longueur totale de 1 440 octets.
- Un objet Output Data doit être invoqué dans un processus d'application d'un appareil E/S.

7.3.2.5 Spécification des services IO Data**7.3.2.5.1 Local Set Input**

Le service Set Input établit de nouvelles données d'entrée d'une baie/sous-baie avec son IOPS pour la prochaine émission pour toutes les relations d'applications connexes. L'IOPS doit être en tout temps cohérent avec les valeurs délivrées d'Input Data. La valeur de l'IOPS

doit seulement être "GOOD" pour des données d'entrée valides de toute la sous-baie. Dans tous les autres cas, la valeur doit être mise à «BAD_BY_xxx» et inclure l'instance détectrice, par exemple sous-baie, baie ou appareil E/S. Ce service est utilisé par le processus d'application de l'appareil E/S pour mettre à jour la partie adressée du tampon cyclique interne de protocole. L'émission réelle du tampon est déclenchée par le temps.

Le service Set Input peut établir l'IOPS seulement. Tel est le cas si la baie/sous/baie ne contient pas de Input Data.

NOTE 1 La mise en œuvre locale de l'interface de service peut prendre en charge des variantes d'accès autres qu'avec la granularité de sous-baie, par exemple pour plus d'une sous-baie ou seulement des parties (variables) de l'élément d'objet Input Data. Cependant, l'élément d'objet Input Data est toujours cohérent avec son IOPS.

Ce service doit seulement être utilisé conjointement à l'IO AR et à la Supervisor AR.

Conformément aux caractéristiques du transport cyclique d'un tampon à l'autre, le comportement suivant est possible:

- Les demandes de "Set Input" sont émises plus vite que le contenu du tampon n'est transporté sur le réseau. Dans ce cas, seules les valeurs fournies avec le plus récent service Set Input sont acheminées sur le réseau.
- Les demandes de "Set Input" sont émises plus lentement que le contenu du tampon n'est transporté sur le réseau. Dans ce cas, les valeurs fournies avec chaque service Set Input sont acheminées plus d'une fois sur le réseau.
- Si les demandes de Set Input sont émises en synchronisme avec le transport du contenu des tampons, les valeurs fournies avec chaque service Set Input sont acheminées une seule fois sur le réseau.

Le Tableau 116 montre les paramètres du service.

Tableau 116 – Local Set Input

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOPS	M	
Subslot Input Data	U	
Result(+)		S
AREP		M
Result (-)		S
AREP		M

NOTE 2 Le service Set Input établit les données d'entrée pour toutes les AR établies qui appartiennent au sous-module. Dans ce cas, les multiples appels pour des entrées partagées ne sont pas nécessaires. Par conséquent, les paramètres de services d'AR et de CR sont omis.

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

IOPS

Ce paramètre doit contenir les valeurs GOOD si les valeurs de données sont valides, BAD_BY_DEVICE, BAD_BY_SLOT, ou BAD_BY_SUBSLOT.

Subslot Input Data

Ce paramètre contient la valeur de l'élément d'objet Input Data.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.2 Local Set Input IOCS

Ce service doit être utilisé par un consommateur (appareil de commande E/S) d'entrées pour passer à la couche application la valeur de l'IOCS modifié. L'IOCS est retourné au fournisseur et fournit un moyen de surveiller l'application. Ce service doit seulement être utilisé conjointement à l'IO AR et à la Supervisor AR.

Par ce service, le processus d'application de l'appareil E/S met à jour le tampon local. Selon les caractéristiques de transport cyclique d'un tampon vers un autre tampon relatives à la relation d'applications utilisée, le comportement suivant est possible:

- Les demandes de Set Input IOCS sont émises plus vite que le contenu des tampons n'est transporté sur le réseau vers le Producteur. Dans ce cas, ce n'est pas chaque valeur de l'IOCS qui est acheminée sur le réseau. Seule la dernière valeur est transportée, les autres sont écrasées localement en écriture.
- Les demandes de Set Input IOCS sont émises plus lentement que le contenu des tampons n'est transporté sur le réseau vers le Producteur. Dans ce cas, chaque valeur de l'IOCS fournie avec le service est acheminée plus d'une fois sur le réseau.
- Si les demandes de Set Input IOCS sont émises en synchronisme avec le transport du contenu des tampons, chaque valeur de l'IOCS fournie avec le service est acheminée sur le réseau.
- S'il y a plusieurs IOCS, seul l'IOCS de la CR non partagée sera indiqué.

Le Tableau 117 montre les paramètres du service.

Tableau 117 – Local Set Input IOCS

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOCS	M	

Nom de paramètre	req	cnf
Result(+)		S
AREP		M
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

IOCS

Ce paramètre doit contenir les valeurs GOOD ou BAD_BY_CONTROLLER.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.3 Local Get Input

Ce service doit seulement être utilisé conjointement à l'IO AR et à la Supervisor AR. Il est utilisé par un appareil de commande E/S pour lire les valeurs d'une sous-baie de données d'entrée avec les IOPS connexes. Le Tableau 118 montre les paramètres du service.

Tableau 118 – Local Get Input

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOPS		M
Subslot Input Data		C
New Flag		M
IOCS		C
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

Result (+)

Ce paramètre indique que la demande de service a réussi.

IOPS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Input Data.

Subslot Input Data

Ce paramètre contient la valeur de l'objet "Input Data".

New Flag

Ce paramètre contient la valeur TRUE si de nouvelles données d'entrée avec IOPS ont été reçues depuis le dernier appel du service. Si le service était appelé plus fréquemment, le fanion New Flag est mis à FALSE par les diagrammes protocolaires.

IOCS

Ce paramètre contient la valeur de l'IOCS au cas où il y aurait un objet Output Data correspondant.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.4 Local Get Input IOCS

Ce service doit seulement être utilisé conjointement à l'IO AR et à la Supervisor AR. Il est utilisé par un fournisseur pour récupérer l'IOCS de l'appareil de commande E/S. Le Tableau 119 montre les paramètres du service.

Tableau 119 – Local Get Input IOCS

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOCS		M
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

Result (+)

Ce paramètre indique que la demande de service a réussi.

IOCS

Ce paramètre contient la valeur de l'IOCS.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.5 Local New Input

Le service New Input signale à l'application de l'appareil de commande E/S une mise à jour du tampon d'entrée après la réception d'une APDU valide. Le Tableau 120 montre les paramètres du service.

Tableau 120 – Local New Input

Nom de paramètre	ind
Argument	M
AREP	M
CREP	M
Slot Number	M
Subslot Number	M
InData Flag	C

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec l'indication.

CREP

Ce paramètre est l'identificateur local pour la CR souhaitée qui est transmis avec l'indication.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

InData Flag

Ce paramètre indique si le tampon reçu est utilisable ou non.

NOTE Tous les IOCR de consommateur de l'AR ont obtenu des données la première fois.

7.3.2.5.6 Local Set Redundancy

Ce service doit être utilisé par l'appareil E/S pour définir le «Redundancy Flag» de l'AR sélectionnée. Le Tableau 121 montre les paramètres du service.

Tableau 121 – Local Set Redundancy

Nom de paramètre	req	cnf
Argument	M	
AREP	U	
Redundancy Flag	U	
Result(+)		S
AREP		M
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec la demande. Il est omis si le paramètre de service Provider State Flag est utilisé, car ce paramètre influence toutes les AR établies transmettant des données E/S.

Redundancy Flag

Ce paramètre indique à la valeur «primary present» qu'une AR au sein de l'ensemble des AR est dans un état «primary». La valeur «no primary present» indique qu'aucune des AR au sein de l'ensemble des AR n'est considérée comme «primary».

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.7 Local Set State

Ce service doit être utilisé par l'appareil E/S pour définir le «Redundancy Flag» de l'AR sélectionnée. Le Tableau 122 montre les paramètres du service.

Tableau 122 – Local Set State

Nom de paramètre	req	cnf
Argument	M	
AREP	U	
State Flag	U	
Result(+)		S
AREP		M
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec la demande. Il est omis si le paramètre de service Provider State Flag est utilisé, car ce paramètre influence toutes les AR établies transmettant des données E/S.

State Flag

Ce paramètre indique si l'AR est en fonctionnement avec la valeur "primary" ou en attente avec la valeur "backup". La valeur par défaut est "primary".

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.8 Local Data State Changed

Le service signale un changement de la valeur de l'APDU Data Status à l'application de l'appareil de commande E/S ou l'appareil E/S. Ce service doit être acheminé seulement si la valeur de l'APDU Data Status a été modifiée par l'appareil E/S ou l'appareil de commande E/S et doit fournir seulement les fanions modifiés. Le Tableau 123 montre les paramètres du service.

Tableau 123 – Local Data State Changed

Nom de paramètre	ind
Argument	M
AREP	M
CREP	M
Data Valid Flag	C
State Flag	C
Redundancy Primary Present Flag	C
Provider State Flag	C
Problem Indicator Flag	C

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec l'indication.

CREP

Ce paramètre est l'identificateur local pour la CR souhaitée qui est transmis avec l'indication.

Data Valid Flag

Ce paramètre indique avec la valeur "DataItem invalid" que toutes les données (y compris IOPS) de l'AR ne sont pas valides et ne doivent pas être traitées par le consommateur.

Le fanion Data Flag avec la valeur «DataItem invalid» pourrait être utilisé si le processus d'application de l'appareil E/S n'est pas capable d'établir Input Data et/ou IOPS en raison d'une erreur fatale d'application. Cependant, il convient de l'utiliser seulement dans des cas sérieux, car le Data Hold Time temporise au niveau du consommateur et met fin à la

connexion. Dans tous les autres cas, il convient que l'IOPS soit utilisé pour informer le consommateur sur le statut des données d'entrée.

La valeur "DataItem valid" indique que les valeurs des données, y compris l'IOPS et l'IOCS, sont valides.

State Flag

Ce paramètre indique si l'AR est en fonctionnement avec la valeur "primary" ou en attente avec la valeur "backup".

Redundancy Primary Present Flag

Ce paramètre indique à la valeur «primary present» qu'une AR au sein de l'ensemble des AR est «primary». La valeur «no primary present» indique qu'aucune des AR au sein de l'ensemble des AR n'est considérée comme «primary».

Provider State Flag

Ce paramètre indique si le processus d'application est en marche avec la valeur "run" ou est arrêté avec la valeur "stop". Cette valeur est donnée à titre informatif seulement pour le processus d'application de consommateur, la validité réelle des données doit être fonction de l'IOPS.

NOTE 1 Ce fanion pourrait être utilisé pour commander un indicateur global pour visualiser le statut global de l'application distante.

Problem Indicator Flag

Ce paramètre indique si au moins une valeur de diagnostic est établie au sein de l'ASE Diagnoses, qui altère l'AR connexe. Cela est indiqué par la valeur "Problem detected". Il doit être mis à "Regular operation" tant qu'il n'existe pas d'entrée de diagnostic.

NOTE 2 Ce fanion pourrait être utilisé pour déclencher la lecture de données de diagnostic au sein de l'appareil de commande E/S. Ce fanion ne dit rien sur la validité des données d'entrée qui est seulement indiquée par l'IOPS. Pour des raisons locales, une entrée de diagnostic peut également faire passer l'IOPS à "BAD" au moyen du service Set Input.

7.3.2.5.9 Read Record Input Data Object Element

Ce service confirmé peut être utilisé pour lire la valeur d'un élément d'objet Input Data. Ce service peut être utilisé conjointement à l'IO AR, à la Supervisor AR ou à l'AR implicite (Implicit AR).

NOTE 1 Une application client peut utiliser le service Read AR Data pour lire les possibles AR UUID des AR établies qui pourraient être utilisées dans ce contexte.

Implicit AR utilisant un Target AR UUID: Ce service contient uniquement un objet Input Data s'il existe une IO AR ou Supervisor AR établie avec le Target AR UUID demandé. Autrement, une erreur de paramètre est donnée comme réponse.

Implicit AR utilisant un NIL AR UUID: Ce service contient seulement un objet Input Data si l'application prend en charge la lecture de valeurs d'objets Input Data sans AR établie. Autrement, une erreur de paramètre est donnée comme réponse.

Ce service est facultatif pour les sous-modules qui ne contiennent aucune donnée d'entrée.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Mais le paramètre Data a une structure et une signification spécifiques à un service. Le Tableau 124 indique les éléments spécifiques de Data pour ce service.

Tableau 124 – Eléments Data du Read Record Input Data Object Element

Nom de paramètre	éléments spécifiques à un service	rsp	cnf
Data (Données)	Length IOCS	M	M(=)
	IOCS	M	M(=)
	Length IOPS	M	M(=)
	IOPS	M	M(=)
	Length Data	M	M(=)
	Data (Données)	M	M(=)

Data

Ce paramètre contient les valeurs de l'objet qui a été lu.

Length IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Input Data.

IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Input Data.

Length IOPS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Input Data.

IOPS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Input Data.

Length Data

Ce paramètre contient la longueur des données de l'objet qui a été lu.

Data

Ce paramètre contient la valeur de l'objet qui a été lu et est constitué du nombre d'octets indiqués dans le paramètre Length Data. Son contenu est composé selon la structure de l'attribut Input Data Structure de l'objet Input Data. Data contient les données d'entrée données par l'appareil E/S.

7.3.2.5.10 Local Set Output

Le service Set Output établit de nouvelles données de sortie d'une baie/sous-baie avec son IOPS pour la prochaine émission pour toutes les relations d'applications connexes. L'IOPS doit être en tout temps cohérent avec les valeurs délivrées d'Output Data. La valeur de l'IOPS doit seulement être "GOOD" pour des données de sortie valides de toute la sous-baie. Dans tous les autres cas, la valeur doit être mise à "BAD_BY_CONTROLLER". Ce service est utilisé par le processus d'application de l'appareil de commande E/S pour mettre à jour la partie adressée du tampon cyclique interne de protocole. L'émission réelle du tampon est déclenchée par le temps.

NOTE La mise en œuvre locale de l'interface de service peut prendre en charge des variantes d'accès autres qu'avec la granularité de sous-baie, par exemple pour plus d'une sous-baie ou seulement des parties (variables) de l'élément d'objet Output Data. Cependant, l'élément d'objet Output Data est toujours cohérent avec son IOPS.

Ce service doit seulement être utilisé conjointement à l'IO AR ou à la Supervisor AR.

Conformément aux caractéristiques du transport cyclique d'un tampon à l'autre, le comportement suivant est possible:

- Les demandes de Set Output sont émises plus vite que le contenu des tampons n'est transporté sur le réseau. Dans ce cas, seules les valeurs fournies avec le plus récent service Set Output sont acheminées sur le réseau.
- Les demandes de Set Output sont émises plus lentement que le contenu des tampons n'est transporté sur le réseau. Dans ce cas, les valeurs fournies avec chaque service Set Output sont acheminées plus d'une fois sur le réseau.

- Si les demandes de Set Output sont émises en synchronisme avec le transport du contenu des tampons, les valeurs fournies avec chaque service Set Output sont acheminées une seule fois sur le réseau.

Le Tableau 125 montre les paramètres du service.

Tableau 125 – Local Set Output

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOPS	M	
Subslot Output Data	M	
Result(+)		S
AREP		M
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

Slot Number

Ce paramètre contient le numéro de la baie.

Subslot Number

Ce paramètre contient le numéro de la sous-baie.

IOPS

Ce paramètre doit contenir les valeurs GOOD si des valeurs de données sont valides ou BAD_BY_CONTROLLER si les données ne sont pas valides.

Subslot Output Data

Ce paramètre contient la valeur de l'élément d'objet Output Data.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.11 Local Set Output IOCS

Ce service doit être utilisé par un consommateur (appareil E/S) de sorties pour passer à la couche application la valeur de l'IOCS modifié. L'IOCS est retourné au fournisseur et fournit un moyen de surveiller l'application. Ce service doit seulement être utilisé conjointement à l'IO AR et à la Supervisor AR.

Par ce service, le processus d'application de l'appareil E/S met à jour le tampon local. Selon les caractéristiques de transport cyclique d'un tampon vers un autre tampon relatives à la relation d'applications utilisée, le comportement suivant est possible:

- Les demandes d'IO Status Change émises plus vite que le contenu des tampons ne sont pas transportées sur le réseau vers le Producteur. Dans ce cas, ce n'est pas chaque valeur de l'IOCS qui est acheminée sur le réseau. Seule la dernière valeur est transportée, les autres sont écrasées localement en écriture.
- Les demandes d'IO Status Change sont émises plus lentement que le contenu des tampons n'est transporté sur le réseau vers le Producteur. Dans ce cas, chaque valeur de l'IOCS fournie avec le service est acheminée plus d'une fois sur le réseau.
- Si les demandes d'IO Status Change sont émises en synchronisme avec le transport du contenu des tampons, chaque valeur de l'IOCS fournie avec le service est acheminée sur le réseau.

Le Tableau 126 montre les paramètres du service.

Tableau 126 – Local Set Output IOCS

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
IOCS	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

CREP

Ce paramètre est l'identificateur local de la CR souhaitée.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

IOCS

Ce paramètre peut contenir les valeurs GOOD, BAD_BY_DEVICE, BAD_BY_SLOT, ou BAD_BY_SUBSLOT.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.12 Local Get Output

Ce service doit être utilisé par un appareil E/S pour lire la valeur d'une baie/sous-baie d'un élément d'objet Output Data conjointement à son IOPS provenant de la couche application. Ce service doit seulement être utilisé conjointement à l'IO AR ou à la Supervisor AR.

L'appareil E/S doit seulement traiter les valeurs de l'élément d'objet Output Data si l'IOPS contient la valeur GOOD. Le Tableau 127 montre les paramètres du service.

Tableau 127 – Local Get Output

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOPS		M
Subslot Output Data		M
New Flag		M
IOCS		C
Result(-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec l'indication.

CREP

Ce paramètre est l'identificateur local pour la CR souhaitée qui est transmis avec l'indication.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

Result (+)

Ce paramètre indique que la demande de service a réussi.

IOPS

Ce paramètre contient la valeur de l'attribut correspondant de l'élément d'objet Output Data.

Subslot Output Data

Ce paramètre contient la valeur de l'objet "Input Data".

New Flag

Ce paramètre contient la valeur TRUE si de nouvelles données de sortie avec IOPS ont été reçues depuis le dernier appel du service. Si le service était appelé plus fréquemment, le fanion New Flag est mis à FALSE par les diagrammes protocolaires.

IOCS

Ce paramètre contient la valeur de l'IOCS au cas où il y aurait un objet Input Data correspondant.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.13 Local Get Output IOCS

Ce service doit être utilisé par un appareil E/S pour lire l'IOCS provenant de la couche application. Ce service doit seulement être utilisé conjointement à l'IO AR ou à la Supervisor AR. Le Tableau 128 montre les paramètres du service.

Tableau 128 – Local Get Output IOCS

Nom de paramètre	req	chf
Argument	M	
AREP	M	
CREP	M	
Slot Number	M	
Subslot Number	M	
Result(+)		S
AREP		M
IOCS		M
Result(-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec l'indication.

CREP

Ce paramètre est l'identificateur local pour la CR souhaitée qui est transmis avec l'indication.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

Result (+)

Ce paramètre indique que la demande de service a réussi.

IOCS

Ce paramètre contient la valeur de l'IOCS.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.14 Local New Output

Le service New Output signale à l'application de l'appareil E/S une mise à jour de la mémoire tampon de sortie après la réception d'une APDU valide. Le Tableau 129 montre les paramètres du service.

Tableau 129 – Local New Output

Nom de paramètre	ind
Argument	M
AREP	M
CREP	M
Slot Number	M
Subslot Number	M
InData Flag	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec l'indication.

CREP

Ce paramètre est l'identificateur local pour la CR souhaitée qui est transmis avec l'indication.

Slot Number

Ce paramètre contient le numéro local de la baie.

Subslot Number

Ce paramètre contient le numéro local de la sous-baie.

InData Flag

Ce paramètre indique si le tampon reçu est utilisable ou non.

NOTE Tous les IOCR de consommateur de l'AR ont obtenu des données la première fois.

7.3.2.5.15 Local Set Provider State

Ce service doit être utilisé par l'appareil de commande E/S pour positionner les fanions «Provider State Flag» dans le champ APDU Data Status de l'APDU véhiculée suivante. Le fanion «Provider State Flag» influence toutes les AR établies de l'appareil E/S. Le Tableau 130 montre les paramètres du service.

Tableau 130 – Local Set Provider State

Nom de paramètre	req	cnf
Argument	M	
AREP	U	
Provider State Flag	U	
Result(+)		S
AREP		M
Result (-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local pour l'AR souhaitée qui est transmis avec la demande. Il est omis si le paramètre de service Provider State Flag est utilisé, car ce paramètre influence toutes les AR établies transmettant des données E/S.

Provider State Flag

Ce paramètre indique si le processus d'application est en marche avec la valeur "run" ou est arrêté avec la valeur "stop". Cette valeur est donnée à titre informatif seulement pour le processus d'application de consommateur, la validité réelle des données doit être fonction de l'IOPS. Dans le cas de "stop", tous les IOPS doivent également être mis à la valeur "BAD_BY_CONTROLLER" au moyen de la demande de service Set Input à toutes les baies/sous-baies.

NOTE Ce fanion pourrait être utilisé pour commander un indicateur global pour visualiser le statut global de l'application distante.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.2.5.16 Read Record Output Data Object Element

Ce service confirmé peut être utilisé pour lire la valeur d'un élément d'objet Output Data avec la valeur substitut. Ce service peut être utilisé conjointement à l'IO AR, à la Supervisor AR ou à l'AR implicite (Implicit AR).

NOTE 1 Une application client peut utiliser le service Read AR Data pour lire les possibles AR UUID des AR établies qui pourraient être utilisées dans ce contexte.

Implicit AR utilisant un Target AR UUID: Ce service contient uniquement un objet Output Data s'il existe une IO AR ou Supervisor AR établie avec le Target AR UUID demandé. Autrement, une erreur de paramètre est donnée comme réponse.

Implicit AR utilisant un NIL AR UUID: Ce service contient seulement un objet Output Data si l'application prend en charge la lecture de valeurs d'objets Output Data sans AR établie. Autrement, une erreur de paramètre est donnée comme réponse.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Mais le paramètre Data a une structure et une

signification spécifiques à un service. Le Tableau 131 indique les éléments spécifiques de Data pour ce service.

Tableau 131 – Éléments Data du Read Record Output Data Object Element

Nom de paramètre	éléments spécifiques à un service	rsp	cnf
Data (Données)	Substitute Active Flag	M	M(=)
	Length IOCS	M	M(=)
	Length IOPS	M	M(=)
	Length Data	M	M(=)
	Data Item	M	M(=)
	IOCS	M	M(=)
	Data (Données)	M	M(=)
	IOPS	M	M(=)
	Substitute Value	M	M(=)
	Substitution Mode	M	M(=)
	Substitute Data Item	M	M(=)
	IOCS	M	M(=)
	Data (Données)	M	M(=)
	Substitute Data Valid	M	M(=)

Data

Ce paramètre contient les valeurs de l'objet qui a été lu.

Substitute Active Flag

Ce paramètre détermine quelles données sont appliquées à la sortie.

Si ce paramètre est mis sur FALSE, le sous-module transporte les valeurs en fonction du Data Item (élément de données).

Si ce paramètre est mis sur TRUE, le sous-module transporte

- les valeurs selon le Substitute Data Item si Substitution Mode est mis sur LAST_VALUE, REPLACEMENT_VALUE ou PROFILE_SPECIFIC,
- les données sont mises à "all zero" («des zéros partout») si Substitution Mode est mis sur ZERO.

Length IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Length IOPS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Length Data

Ce paramètre contient la longueur des données de l'objet qui a été lu.

Data Item

Ce paramètre est constitué des éléments suivants.

IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Data

Ce paramètre contient la valeur de l'objet qui a été lue et est constitué du nombre d'octets indiqués dans le paramètre Length Data de la réponse. Son contenu est composé selon la structure de l'attribut Output Data Structure de l'objet Output Data. Data contient les données de sortie données par l'appareil de commande E/S.

IOPS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Substitute Value

Ce paramètre est constitué des éléments suivants.

Substitution Mode

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Substitute Data Item

Ce paramètre est constitué des éléments suivants.

IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Data

Ce paramètre contient la valeur de l'objet qui a été lue et est constitué du nombre d'octets indiqués dans le paramètre Length Data de la réponse. Son contenu est composé selon la structure de l'attribut Output Data Object Format de l'objet Output Data. Data contient les données de remplacement données par l'appareil de commande E/S.

Substitute Data Valid

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

7.3.2.5.17 Read Substitute Value

Ce service confirmé peut être utilisé pour lire la valeur substitut. Ce service peut être utilisé conjointement à l'IO AR, à la Supervisor AR ou à l'AR implicite (Implicit AR).

NOTE 1 Une application client peut utiliser le service Read AR Data pour lire les possibles AR UUID des AR établies qui pourraient être utilisées dans ce contexte.

Implicit AR utilisant un Target AR UUID: Ce service contient uniquement un objet Substitute Value s'il existe une IO AR ou Supervisor AR établie avec le Target AR UUID demandé. Autrement, une erreur de paramètre est donnée comme réponse.

Implicit AR utilisant l'API: Ce service contient seulement un objet Substitute Value si l'application prend en charge la lecture de valeurs d'objets Substitute Value sans AR établie. Autrement, une erreur de paramètre est donnée comme réponse.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Mais le paramètre Data a une structure et une signification spécifiques à un service. Le Tableau 132 indique les éléments spécifiques de Data pour ce service.

Tableau 132 – Eléments Data du Read Substitute Value

Nom de paramètre	éléments spécifiques à un service	rsp	cnf
Data (Données)	Substitute Value	M	M(=)
	Substitution Mode	M	M(=)
	Substitute Data Item	M	M(=)
	IOCS	M	M(=)
	Data (Données)	M	M(=)
	Substitute Data Valid	M	M(=)

Data

Ce paramètre contient les valeurs de l'objet qui a été lu.

Substitute Value

Ce paramètre est constitué des éléments suivants.

Substitution Mode

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Substitute Data Item

Ce paramètre est constitué des éléments suivants.

IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Data

Ce paramètre contient la Substitute Value qui a été lue et est constitué du nombre d'octets indiqués dans le paramètre Length Data de la réponse. Son contenu est composé selon la structure de l'attribut Output Data Object Format de l'objet Output Data.

Substitute Data Valid

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Si le sous-module ne prend pas en charge Substitute Value, une réponse négative contenant les valeurs Error Decode = "PNIORW", Error code 1 = "invalid index", Error code 2 = "user specific" peuvent être données ou bien un enregistrement vide peut être renvoyé.

7.3.2.5.18 Write Substitute Value

Ce service confirmé peut être utilisé pour écrire la valeur substitut d'un élément d'objet Output Data. Ce service peut être utilisé conjointement à l'IO AR et à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service d'écriture (voir 7.3.1.4.2.1). Mais le paramètre Data a une structure et une signification spécifiques à un service. Le Tableau 133 indique les éléments spécifiques de Data pour ce service.

Tableau 133 – Eléments Data du Write Substitute Value

Nom de paramètre	éléments spécifiques à un service	req	ind
Data (Données)	Substitute Value	M	M(=)
	Substitution Mode	M	M(=)
	Substitute Data Item	M	M(=)
	IOCS	M	M(=)
	Data (Données)	M	M(=)
	Substitute Data Valid	M	M(=)

Data

Ce paramètre contient la valeur de l'objet qui peut être utilisée pour écrire.

Substitute Value

Ce paramètre est constitué des éléments suivants.

Substitution Mode

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Substitute Data Item

Ce paramètre est constitué des éléments suivants.

IOCS

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

Data

Ce paramètre contient la Substitute Data et est constitué du nombre d'octets indiqués dans le paramètre Length Data de la réponse. Son contenu est composé selon la structure de l'attribut Output Data Object Format de l'objet Output Data.

Substitute Data Valid

Ce paramètre contient la valeur de l'attribut correspondant de l'objet Output Data.

7.3.2.6 Comportement des objets IO Data

7.3.2.6.1 Comportement général de l'objet Input Data

Pour des raisons de performance, la valeur de l'objet Input Data doit être transférée à la suite d'un changement le plus rapidement possible au tampon des IO AR en vue de l'émission. La vitesse et les giges de cette action constituent un paramètre de performance d'un appareil E/S.

7.3.2.6.2 Comportement général de l'objet Output Data

Pour des raisons de performance, la valeur pour l'objet Output Data doit être transférée dès la réception le plus rapidement possible à l'objet. La vitesse et les giges de cette action constituent un paramètre de performance d'un appareil E/S. Ce paramètre détermine l'exactitude de la synchronisation.

Le service Read Output doit fournir la valeur courante du processus d'application qui a été récupérée avec le service Get Output.

NOTE Pour des raisons différentes (retards de sortie physique, par exemple), cette valeur peut différer de la valeur des données de sortie réelles ou de la dernière valeur transférée.

7.3.3 ASE LogBook Data

7.3.3.1 Aperçu général

L'ASE LogBook Data fournit un moyen de stocker des événements générés localement et spécifiques à une AR dans un tampon circulaire et fournit un service pour lire ce tampon d'événements. Le contenu des entrées est spécifique à un fabricant et peut être utilisé pour identifier des problèmes de communications, par exemple les causes d'arrêt d'une AR.

L'ASE LogBook Data définit des attributs des objets LogBook Data et fournit un jeu de services utilisés pour lire à distance et écrire localement leurs valeurs. Un ASE LogBook Data est spécifique pour un appareil E/S. L'objet LogBook Data est adressé de façon implicite par les services définis.

L'accès à des objets LogBook Data est réalisé conformément au modèle d'accès Client/Serveur pour la lecture. Le modèle Client/Serveur est caractérisé en ce que l'application Client envoie une demande de lecture à une application Serveur qui répond en conséquence.

Le modèle formel de l'ASE LogBook Data est présenté par la spécification pour la classe LogBook Data, contenant une description de ses attributs, de ses services ainsi que ses invocations, suivie d'une spécification particulière des services.

7.3.3.2 Spécification de la classe LogBook Data

7.3.3.2.1 Modèle

Un objet LogBook Data est décrit par le modèle suivant:

ASE: ASE LogBook Data

CLASS: LogBook Data

CLASS ID: pas utilisé

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Attribut-clé:	Implicit
2	(m) Attribut:	Current Local Time Stamp
3	(m) Attribut:	List of Entries
3.1	(m) Attribut:	Local Time Stamp

- 3.2 (m) Attribut: AR UUID
 3.3 (m) Attribut: PNIO Status
 3.4 (m) Attribut: Entry Detail

SERVICES:

- 1 (m) OpsService: Read LogBook Data
 2 (m) OpsService: Local Create LogBook Entry

7.3.3.2.2 Attributs**Implicit**

L'attribut clé « Implicit » indique que l'objet LogBook Data est adressé de façon implicite par les services.

Current Local Time Stamp

Cet attribut contient l'heure locale courante au moyen de la valeur du compteur de cycles.

Type d'attribut: Unsigned64

List of Entries

Un appareil E/S doit prendre en charge au moins 16 entrées LogBook et un appareil de commande E/S doit prendre en charge au moins 4 Ko. Cette liste d'attributs est constituée des attributs suivants:

Local Time Stamp

Cet attribut contient l'heure locale à laquelle l'entrée a été effectuée au moyen de la valeur du compteur de cycles à cet instant.

Type d'attribut: Unsigned64

AR UUID

Cet attribut définit l'UUID de l'entrée IO AR qui est la source.

Type d'attribut: UUID

PNIO Status

Cet attribut contient la valeur définie de diagramme protocolaire pour l'entrée.

Type d'attribut: Unsigned32

Valeurs permises: voir CEI 61158-6-10

Entry Detail

Cet attribut contient la valeur définie de diagramme protocolaire pour l'entrée.

Type d'attribut: Unsigned32

7.3.3.3 Spécification des services LogBook Data**7.3.3.3.1 Read LogBook Data**

Ce service confirmé peut être utilisé pour lire la valeur d'un objet LogBook Data. Ce service peut être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

Si le journal réel contient plus d'octets que la longueur demandée, seules les entrées les plus récentes (en fonction de la quantité d'entrées demandées) doivent être utilisées pour la réponse. La liste retournée n'a pas besoin d'être triée rapidement; le demandeur de l'information est capable de faire cela grâce aux horodateurs locaux.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Mais le paramètre Data a une structure et une signification spécifiques à un service. Le Tableau 134 indique les éléments spécifiques de Data pour ce service.

Tableau 134 – Eléments Data de Read LogBook Data

Nom de paramètre	éléments spécifiques à un service	rsp	cnf
Data (Données)	Current Local Time Stamp	M	M(=)
	List of Entries	M	M(=)
	Local Time Stamp	M	M(=)
	AR UUID	M	M(=)
	PNIO Status	M	M(=)
	Entry Detail	M	M(=)

Data

Ce paramètre contient les valeurs de l'objet qui a été lu.

Current Local Time Stamp

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

List of Entries

Ce paramètre est constitué des éléments de liste suivants:

Local Time Stamp

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

AR UUID

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

PNIO Status

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Entry Detail

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

7.3.3.3.2 Local Create LogBook Entry

Ce service local indique la valeur de l'objet LogBook Data. Le Tableau 135 montre les paramètres du service.

Tableau 135 – Local Create LogBook Entry

Nom de paramètre	ind
Argument	M
AREP	M
Local Time Stamp	M
AR UUID	M
PNIO Status	M
Entry Detail	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Local Time Stamp

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

AR UUID

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

PNIO Status

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Entry Detail

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

7.3.4 ASE Diagnosis**7.3.4.1 Vue d'ensemble**

Dans l'environnement de la couche d'application, l'ASE Diagnosis d'un appareil E/S contient différentes entrées de diagnostic (Diagnosis) qui sont utilisées pour conserver les informations de diagnostic de l'appareil E/S. Les informations de diagnostic issues de l'appareil E/S sont relatives à l'AP adressé et incluent seulement les baies, sous-baies et voies qui ont établi un diagnostic.

NOTE Si une sous-baie ou une voie n'a pas de diagnostic établi, ces informations ne sont pas émises.

L'ASE Diagnosis définit les attributs des objets Diagnosis et fournit un ensemble de services utilisés pour lire leurs valeurs. Les types de diagnostic suivants sont définis:

- Channel Diagnosis
- Extended Channel Diagnosis
- Qualified Channel Diagnosis (Diagnostic qualifié de voie)
- Manufacturer Specific Diagnosis

Il est fortement recommandé d'utiliser Channel Diagnosis, Ext Channel Diagnosis ou Qualified Channel Diagnosis dans tous les cas concordants, car la sémantique détaillée est totalement normalisée. Uniquement lorsque ce qui précède ne convient pas, les données de diagnostic spécifiques au fabricant ne peuvent être acheminées au moyen du Manufacturer Specific Diagnosis.

Notez que le diagnostic de voie, le diagnostic de voie étendu et le diagnostic de voie qualifié diffèrent seulement par la quantité de données fournies par l'appareil E/S pour l'élément de diagnostic spécifique. Le diagnostic de voie offre des propriétés de voie et le type d'erreur de voie. Le diagnostic de voie étendu ajoute un type d'erreur de voie étendu et une valeur supplémentaire. Le diagnostic de voie qualifié ajoute le qualificatif de voie qualifié. Le format choisi pour une entrée de diagnostic particulière est décidé par le fabricant de l'appareil E/S. Ainsi, le même format peut être tout le temps choisi (par ex. utiliser toujours le format de diagnostic de voie étendu).

Le diagnostic spécifique au fabricant fournit les propriétés de la voie, un identificateur de structure d'utilisateur (pour préciser le contenu des données suivantes) et des données brutes.

La sévérité est définie sous forme de diagnostic, de maintenance exigée, maintenance requise et statut. La classification du diagnostic, de la maintenance et qualification en fonction de leur sévérité est indiquée à la Figure 27.

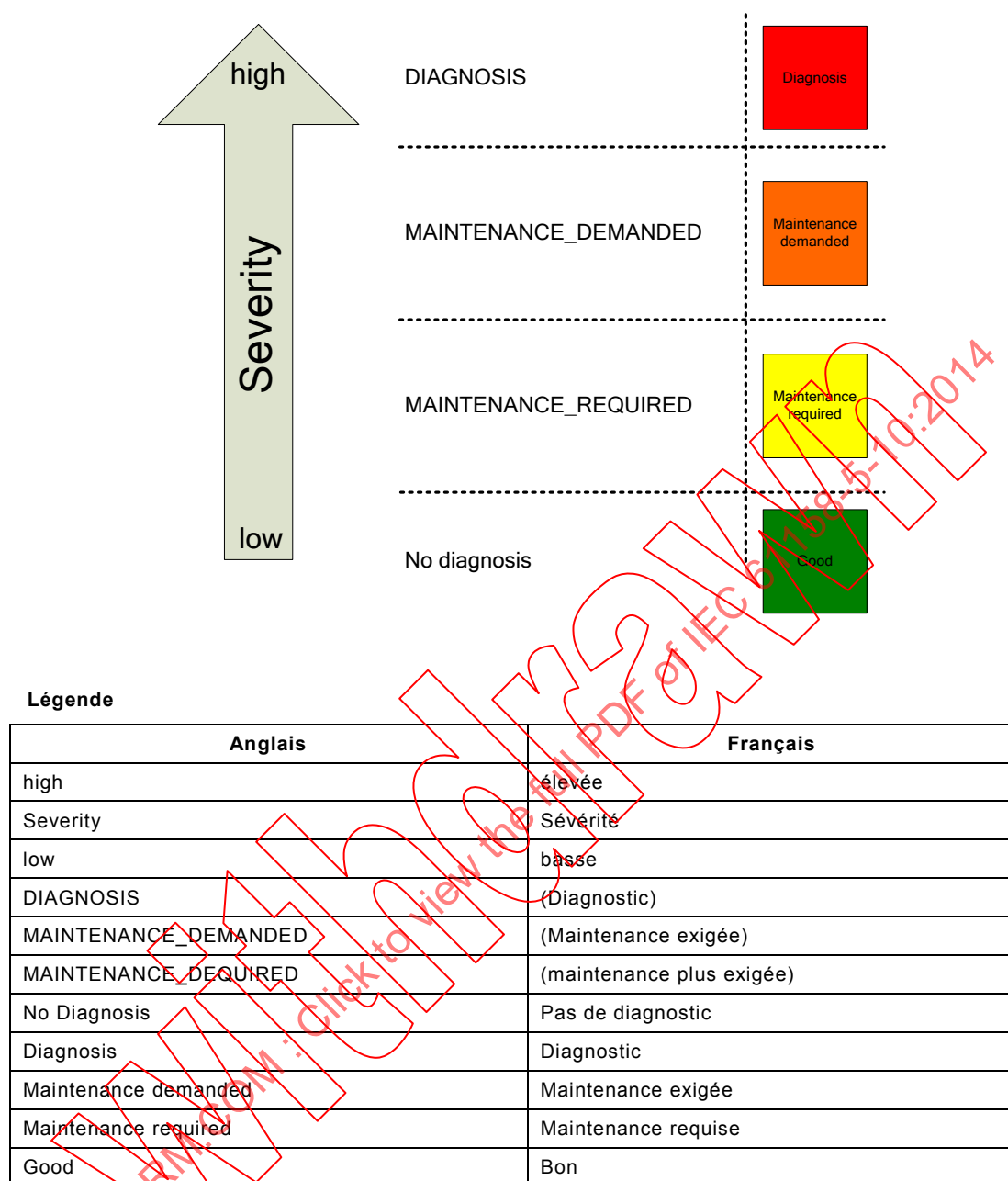


Figure 27 – Classification de la sévérité du diagnostic et de la maintenance

La sévérité d'un diagnostic de voie est exprimée par l'attribut ChannelProperties.Maintenance.

Si l'attribut est mis sur DIAGNOSIS, MAINTENANCE_DEMANDED, ou MAINTENANCE_REQUIRED, la sévérité est définie selon la valeur de l'attribut (voir Figure 27).

NOTE L'appareil E/S et l'appareil de commande E/S peuvent déduire la signalisation de l'utilisateur (par ex. par des DEL) à partir de cette sévérité (voir Figure 27).

Si l'attribut est mis sur la valeur QUALIFIED_DIAGNOSIS, la signification et la sévérité sont déduites de QualifiedChannelQualifier par des applications. Les qualificatifs 2 à 31 dans cet attribut sont classés par un profil. Ainsi, la valeur QUALIFIED_DIAGNOSIS n'est pas considérée comme étant sévère par l'appareil E/S et l'appareil de commande E/S. Le niveau de diagnostic le plus élevé (par ex. la station de maintenance) peut déduire une sévérité à partir des qualificatifs.

Le «status» de sévérité n'est fourni que pour le diagnostic spécifique au fabricant. Les Manufacturer Specific Diagnosis indiquant diagnostic, maintenance exigée et maintenance requise ont des ChannelProperties.Specifier définies sur APPEARS. Au contraire un Manufacturer Specific Diagnosis indiquant un statut a une ChannelProperties.Maintenance définie sur DIAGNOSIS et ChannelProperties.Specifier set sur DISAPPEARS.

L'application d'un appareil E/S peut fournir la valeur d'un ou plusieurs objets Diagnosis à l'AR implicite, à l'IO AR et à la Supervisor AR.

Le modèle de Diagnosis utilise le modèle d'accès Client/Serveur.

De plus, en cas d'établissement de la première entrée de diagnostic, l'ASE Input Data doit en être informé au moyen de l'événement (Diagnosis ASE "First Diagnosis Entry set"). De plus, en cas de suppression de la dernière entrée de diagnostic, l'ASE Input Data doit en être informé au moyen de l'événement (Diagnosis ASE "Last Diagnosis Entry cleared"). L'ASE Input Data ne doit pas être informé si seules des informations de maintenance sont établies ou supprimées.

L'ASE Alarm peut également être informé afin d'envoyer ces types d'alarme définies dans le Tableau 161 et avec la mention «Attached to Diagnosis ASE = Yes» si elle est configurée par l'application.

7.3.4.2 Spécification de la classe Diagnosis

7.3.4.2.1 Modèle

L'objet Diagnosis est décrit par le modèle suivant:

ASE:	ASE Diagnosis
CLASS:	Diagnosis (Diagnostic)
CLASS ID:	pas utilisé
PARENT CLASS:	TOP
ATTRIBUTS:	
1	(m) Attribut-clé: Implicit
2	(m) Attribut: Device Status
3	(c) Contrainte: Device Status <> DEVICE_OK
3.1	(m) Attribut: List of APs
3.1.1	(m) Attribut: API
3.1.2	(m) Attribut: List of Slots
3.1.2.1	(m) Attribut: Slot Number
3.1.2.2	(m) Attribut: List of Subslots
3.1.2.2.1	(m) Attribut: Subslot Number
3.1.2.2.2	(m) Attribut: Submodule Diagnosis State
3.1.2.2.3	(m) Attribut: List of Channels
3.1.2.2.3.1	(m) Attribut: Channel Number
3.1.2.2.3.2	(m) Attribut: List of Channel Diagnosis
3.1.2.2.3.2.1	(m) Attribut: Time Stamp
3.1.2.2.3.2.2	(m) Attribut: Channel Properties
3.1.2.2.3.2.2.1	(m) Attribut: Type
3.1.2.2.3.2.2.2	(m) Attribut: Accumulative
3.1.2.2.3.2.2.3	(m) Attribut: Maintenance
3.1.2.2.3.2.2.4	(m) Attribut: Specifier (spécificateur)
3.1.2.2.3.2.2.5	(m) Attribut: Direction, sens
3.1.2.2.3.2.3	(m) Attribut: User Structure Identifier
3.1.2.2.3.2.4	(o) Attribut: Channel Error type
3.1.2.2.3.2.5	(o) Attribut: Ext Channel Error type

3.1.2.2.3.2.6	(o) Attribut:	Ext Channel Add Value
3.1.2.2.3.2.7	(o) Attribut:	Qualified Channel Qualifier
3.1.2.2.3.2.8	(o) Attribut:	List of Manufacturer Specific Diagnosis
3.1.2.2.3.2.8.1	(m) Attribut:	Data (Données)
3.2	(o) Attribute:	List of Nested Diagnosis Info
3.2.1	(m) Attribute:	IP Address
3.2.2	(m) Attribute:	Station Name Length
3.2.3	(m) Attribute:	Name Of Station Value

SERVICES:

1	(m) OpsService:	Local Add Diagnosis Entry
2	(m) OpsService:	Local Remove Diagnosis Entry
3	(m) OpsService:	Local Diagnosis Event
4	(m) OpsService:	Read Diagnosis in channel coding for one subslot
5	(m) OpsService:	Read Diagnosis in all codings for one subslot
6	(m) OpsService:	Read Maintenance required in channel coding for one subslot
7	(m) OpsService:	Read Maintenance required in all codings for one subslot
8	(m) OpsService:	Read Maintenance demanded in channel coding for one subslot
9	(m) OpsService:	Read Maintenance demanded in all codings for one subslot
10 subslot	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one subslot
11	(m) OpsService:	Read Diagnosis in channel coding for one slot
12	(m) OpsService:	Read Diagnosis in all codings for one slot
13	(m) OpsService:	Read Maintenance required in channel coding for one slot
14	(m) OpsService:	Read Maintenance required in all codings for one slot
15	(m) OpsService:	Read Maintenance demanded in channel coding for one slot
16	(m) OpsService:	Read Maintenance demanded in all codings for one slot
17	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one slot
18	(m) OpsService:	Read Diagnosis in channel coding for one AR
19	(m) OpsService:	Read Diagnosis in all codings for one AR
20	(m) OpsService:	Read Maintenance required in channel coding for one AR
21	(m) OpsService:	Read Maintenance required in all codings for one AR
22	(m) OpsService:	Read Maintenance demanded in channel coding for one AR
23	(m) OpsService:	Read Maintenance demanded in all codings for one AR
24	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one AR
25	(m) OpsService:	Read Diagnosis in channel coding for one API
26	(m) OpsService:	Read Diagnosis in all codings for one API
27	(m) OpsService:	Read Maintenance required in channel coding for one API
28	(m) OpsService:	Read Maintenance required in all codings for one API
29	(m) OpsService:	Read Maintenance demanded in channel coding for one API
30	(m) OpsService:	Read Maintenance demanded in all codings for one API
31	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one API
33 device	(m) OpsService:	Read Diagnosis, Maintenance, Qualified and Status for one device
34	(o) OpsService:	Read Nested Diagnosis Info

7.3.4.2.2 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet Diagnosis est adressé de façon implicite par les services.

Device Status

Cet attribut contient le résumé de diagnostic. Si la valeur est mise à DEVICE_OK, aucune baie/sous-baie ne doit contenir d'informations de diagnostic. Si la valeur est mise à DEVICE_FAILURE, MAINTENANCE, ou MANUFACTURER_SPECIFIC_STATUS, au moins une baie/sous-baie doit contenir des informations de diagnostic, des informations de maintenance ou des informations de statut spécifiques à un fabricant.

Type d'attribut: Unsigned8

Valeurs permises: DEVICE_OK, DEVICE_FAILURE, MAINTENANCE ou MANUFACTURER_SPECIFIC_STATUS

List of APs

Cet attribut contient les informations de diagnostic, de maintenance, qualifiées ou de statut spécifiques à un fabricant pour chaque processus d'application pris en charge. Un élément de liste est constitué des attributs suivants:

API

Cet attribut doit être seulement constitué des API avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de statut spécifiques à un fabricant.

List of Slots

Cet attribut contient seulement des baies avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de statut spécifiques à un fabricant. Un élément de liste est constitué des attributs suivants:

Slot Number

Cet attribut définit le numéro de baie comme un identificateur local. Ce numéro est unique au sein d'un appareil. La numérotation des baies au sein d'un appareil doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. La liste doit contenir toutes les baies physiques ou logiques avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de statut spécifiques à un fabricant.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

List of Subslots

Cet attribut contient seulement des sous-baies avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de statut spécifiques à un fabricant. Un élément de liste est constitué des attributs suivants:

Subslot Number

Cet attribut définit le numéro de sous-baie comme un identificateur local. Ce numéro est unique au sein d'une baie. La numérotation des sous-baies au sein d'une baie doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. La liste doit contenir toutes les sous-baies physiques ou logiques avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de statut spécifiques à un fabricant.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

NOTE 1 La valeur de Subslot Number 0 est utilisée conjointement à AlarmType «Pull» pour adresser un certain module dans l'appareil E/S; elle ne contient jamais un diagnostic.

Submodule Diagnosis State

Cet attribut contient le résumé des diagnostics pour le sous-module. La valeur SUBMODULE_FAILURE doit seulement être mise en cas d'au moins une sévérité de Channel Diagnosis autre que INFORMATION et au moins un Manufacturer Specific Diagnosis avec une sévérité autre que INFORMATION et le spécificateur de propriétés de voie ("channel properties specifier") doit être mis à APPEARS. Dans tous les autres cas, la valeur doit être mise à SUBMODULE_OK.

Type d'attribut: Unsigned8

Valeurs permises: SUBMODULE_OK, SUBMODULE_FAILURE

List of Channels

Cet attribut contient seulement des voies avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de

statut spécifiques à un fabricant. Un élément de liste est constitué des attributs suivants:

Channel Number

Cet attribut définit le numéro de voie comme un identificateur local. Ce numéro est unique au sein d'une sous-baie. La numérotation des voies au sein d'une sous-baie doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. La liste doit contenir toutes les voies physiques ou logiques avec des informations de diagnostic, des informations de maintenance, des informations qualifiées ou des informations de statut spécifiques à un fabricant.

La valeur spéciale 0x8000 identifie l'ensemble du sous-module, au lieu d'une voie particulière.

Dans ces cas, l'attribut Channel Properties Accumulative doit contenir la valeur INDIVIDUAL.

Si l'attribut Channel Properties Accumulative contient la valeur CHANNELGROUP, cet attribut contient le numéro de la voie affectée la plus basse.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF numéro de voie spécifique à un utilisateur, 0x8000 sous-module

List of Channel Diagnosis

Cette liste contient les informations de diagnostic, de maintenance, qualifiées ou de diagnostic ou de statut spécifiques à un fabricant, liées à cette voie. Un élément de liste est constitué des attributs suivants:

Time Stamp

Cet attribut contient l'heure à laquelle les informations de diagnostic, de maintenance, qualifiées ou de diagnostic ou de statut spécifiques à un fabricant.

Type d'attribut: TimeStamp

Channel Properties

Cet attribut est constitué des attributs suivants:

Type

Cet attribut indique la longueur des données de la voie à laquelle l'objet Channel Diagnosis se rapporte.

Valeurs permises:

1BIT, 2BIT, 4BIT, BYTE, WORD, 2WORDS, 4WORDS, UNSPECIFIC

La valeur UNSPECIFIC doit être utilisée si le Channel Number contient la valeur 0x8000 ou si la longueur de données est différente de toutes les autres valeurs autorisées.

Accumulative

Cet attribut indique si le diagnostic est relatif à une certaine voie ou à un groupe de voies.

Valeurs permises: INDIVIDUAL, CHANNELGROUP

Maintenance

Cet attribut indique la sévérité du diagnostic.

La valeur DIAGNOSIS est utilisée pour indiquer le diagnostic, y compris le diagnostic et le statut spécifiques au fabricant.

Les valeurs MAINTENANCE_REQUIRED et MAINTENANCE_DEMAN_DED sont utilisées pour indiquer l'état de la maintenance.

NOTE 2 Par exemple, une imprimante indique «niveau bas du toner» comme exigence de maintenance (par exemple, pour commander une nouvelle cartouche de toner) et «niveau bas du toner, qualité d'impression affectée» comme exigence de maintenance (par exemple pour remplacer la cartouche de toner).

La valeur QUALIFIED_DIAGNOSIS est utilisée pour indiquer le diagnostic qualifié.

Valeurs permises: DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS

Spécifier (spécificateur)

Cet attribut indique le spécificateur de diagnostic.

Valeurs permises:

APPEARS – pour diagnostic, maintenance, diagnostic spécifique au fabricant ou qualifié

DISAPPEARS – pour informations de statut spécifique à un fabricant

NOTE 3 Par exemple, une imprimante indique «toner vide, imprimante arrêtée» comme diagnostic «appears».

Sens

Cet attribut indique le sens de la voie à laquelle l'objet Channel Diagnosis se rapporte.

Type d'attribut: Unsigned8

Valeurs permises: INPUT_CHANNEL, OUTPUT_CHANNEL, BIDIRECTIONAL_CHANNEL, MANUFACTURER_SPECIFIC

User Structure Identifier

Ce paramètre indique le format des données autorisées.

Type d'attribut: Unsigned16

Valeurs permises: CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, QUALIFIED_CHANNEL_DIAGNOSIS_USI, MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF

NOTE 4 L'appareil E/S détermine dans quel format un élément de diagnostic spécifique est renvoyé. Le demandeur peut déterminer si les données spécifiques à un fabricant doivent être retournées (en demandant «tous les codages») ou non (en demandant le «codage de voie»).

Channel Error type

Cet attribut facultatif indique le type d'erreur de l'objet Channel Related Diagnosis. Il n'est présent que si l'attribut User Structure Identifier est mis sur CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Type d'attribut: Unsigned16

Tableau 136 – Channel Error Type

Channel Error Type	Signification
SHORT_CIRCUIT	Court-circuit
UNDERVOLTAGE	Sous-tension
OVERVOLTAGE	Surtension
OVERLOAD	Surcharge
OVERTEMPERATURE	Sur-température
LINE_BREAK	Rupture de ligne
UPPER_LIMIT_VALUE_EXCEEDED	Valeur limite supérieure dépassée
LOWER_LIMIT_VALUE_EXCEEDED	Valeur limite inférieure dépassée
ERROR	Erreur
SIMULATION_ACTIVE	Simulation active
PARAMETER_MISSING	Paramètre manquant
PARAMETERIZATION_FAULT	Défaut de paramétrage
POWER_SUPPLY_FAULT	Défaut d'alimentation

Channel Error Type	Signification
FUSE_BLOWN_OR_OPEN	Fusible fondu/ouvert
COMMUNICATION_FAULT	Défaut de communication
GROUND_FAULT	Défaut à la terre
REFERENCE_POINT_LOST	Point de référence perdu
PROCESS_EVENT_LOST	Événement de processus perdu/erreur d'échantillonnage
THRESHOLD_WARNING	Avertissement de seuil
OUTPUT_DISABLED	Sortie désactivée
SAFETY_EVENT	Événement de sécurité
EXTERNAL_FAULT	Défaut externe
TEMPORARY_FAULT	Défaut temporaire
DATA_TRANSMISSION_IMPOSSIBLE	Transmission de données impossible
REMOTE_MISMATCH	Discordance du système distant
MEDIA_REDUNDANCY_MISMATCH	Discordance de la redondance de supports
SYNC_MISMATCH	Discordance de synchronisation
ISOCRONOUS_MODE_MISMATCH	Discordance du mode isochrone
MULTICAST_CR_MISMATCH	Discordance de la CR multidiffusion
FIBER_OPTIC_MISMATCH	Diagnostic de fibre optique
NETWORK_COMPONENT_FUNCTION_MISMATCH	Discordance de la fonction de composant réseau
TIME_MISMATCH	Discordance de l'heure
DYNAMIC_FRAME_PACKING_FUNCTION_MISMATCH	Discordance de la fonction de condensation de trame dynamique
MEDIA_REDUNDANCY_WITH_PLANNED_DUPLICATION_MISMATCH	Discordance de la redondance de supports avec duplication planifiée
SYSTEM_REDUNDANCY_MISMATCH	Discordance de la redondance du système
MULTIPLE_INTERFACE_MISMATCH	Discordance des multiples interfaces
NESTED_DIAGNOSIS_INDICATION	Indication de diagnostic imbriquée
MANUFACTURER_SPECIFIC_001B – MANUFACTURER_SPECIFIC_001E, MANUFACTURER_SPECIFIC_0100 – MANUFACTURER_SPECIFIC_7FFF	Propres au fabricant
PROFILE_SPECIFIC_0020 à PROFILE_SPECIFIC_00FF	Profils communs
PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF	Profils

Ext Channel Error type

Cet attribut facultatif dépend de l'attribut «Channel Error type» et indique un type d'erreur étendu, le cas échéant. Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI. Les valeurs admissibles sont montrées dans le Tableau 137.

Type d'attribut: Unsigned16

Tableau 137 – Ext Channel Error type

Channel Error Type	Ext Channel Error Type
SHORT_CIRCUIT	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
UNDERVOLTAGE	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
OVERVOLTAGE	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
OVERLOAD	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
OVERTEMPERATURE	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
LINE_BREAK	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
UPPER_LIMIT_VALUE_EXCEEDED	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
LOWER_LIMIT_VALUE_EXCEEDED	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
ERROR	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF

Channel Error Type	Ext Channel Error Type
PARAMETER_MISSING	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
SIMULATION_ACTIVE	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
PARAMETERIZATION_FAULT	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
POWER_SUPPLY_FAULT	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
FUSE_BLOWN_OR_OPEN	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
COMMUNICATION_FAULT	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
GROUND_FAULT	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
REFERENCE_POINT_LOST	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
PROCESS_EVENT_LOST	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
THRESHOLD_WARNING	MANUFACTURER_SPECIFIC_0001 à
	MANUFACTURER_SPECIFIC_7FFF

Channel Error Type	Ext Channel Error Type
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
OUTPUT_DISABLED	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
SAFETY_EVENT	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
EXTERNAL_FAULT	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
TEMPORARY_FAULT	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	ACCUMULATIVE_INFO
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
DATA_TRANSMISSION_IMPOSSIBLE	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	LINK_STATE_MISMATCH_LINK_DOWN
	MAU_TYPE_MISMATCH
	LINE_DELAY_MISMATCH
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
REMOTE_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	PEER_CHASSIS_ID_MISMATCH
	PEER_PORT_ID_MISMATCH
	PEER_RT_CLASS_3_MISMATCH
	PEER_MAUTYPE_MISMATCH
	PEER_MRP_DOMAIN_MISMATCH
	NO_PEER_DETECTED
	PEER_LINE_DELAY_MISMATCH
	PEER_PTCP_MISMATCH
	PEER_PREAMBLE_LENGTH_MISMATCH
	PEER_FRAGMENTATION_MISMATCH
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
MEDIA_REDUNDANCY_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF

Channel Error Type	Ext Channel Error Type
	MANAGER_ROLE_FAIL
	MRP_RING_OPEN
	MULTIPLE_MANAGER
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
SYNC_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	NO_SYNC_MESSAGE_RECEIVED
	JITTER_OUT_OF_BOUNDARY
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
ISOCRONOUS_MODE_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	OUTPUT_TIME_FAILURE
	INPUT_TIME_FAILURE
	MASTER_LIFE_SIGN_FAILURE
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
MULTICAST_CR_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	MULTICAST_CONSUMER_CR_TIMED_OUT
	ADDRESS_RESOLUTION_FAILED
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
FIBER_OPTIC_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	POWER_BUDGET
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
NETWORK_COMPONENT_FUNCTION_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	FRAME_DROPPED_NO_RESOURCE
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
DYNAMIC_FRAME_PACKING_FUNCTION_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	FRAME_LATE_ERROR_FOR_FRAME_0100 à FRAME_LATE_ERROR_FOR_FRAME_0FFF
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
MEDIA_REDUNDANCY_WITH_PLANNED_DUPLICATION_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	MRPD_DUPLICATION_VOID_FOR_FRAME_0700 à MRPD_DUPLICATION_VOID_FOR_FRAME_0FFF
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
SYSTEM_REDUNDANCY_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF

Channel Error Type	Ext Channel Error Type
	SYSTEM_REDUNDANCY_EVENT
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
MULTIPLE_INTERFACE_MISMATCH	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	STANDARD_GATEWAY_MISMATCH
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
NESTED_DIAGNOSIS_INDICATION	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	SUBORDINATED_ENTITY_INFORMATION
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF
MANUFACTURER_SPECIFIC_001B – MANUFACTURER_SPECIFIC_001E, MANUFACTURER_SPECIFIC_0100 – MANUFACTURER_SPECIFIC_7FFF	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_90FF	MANUFACTURER_SPECIFIC_0001 à MANUFACTURER_SPECIFIC_7FFF
	PROFILE_SPECIFIC_9000 à PROFILE_SPECIFIC_9FFF

Les combinaisons autorisées de Channel Error Type et Ext Channel Error Type sont présentées dans le Tableau 138.

Tableau 138 – Combinaisons autorisées de Channel Error Type et Ext Channel Error Type

ChannelErrorType	ExtChannelErrorType	Utilisation
Normatif	Normatif	Autorisée
Normatif	Spécifique à un vendeur	Autorisée
Normatif	Spécifique à un profil	Autorisée
Spécifique à un vendeur	Normatif	Interdite
Spécifique à un vendeur	Spécifique à un vendeur	Autorisée
Spécifique à un vendeur	Spécifique à un profil	Interdite
Spécifique à un profil	Normatif	Interdite
Spécifique à un profil	Spécifique à un vendeur	Autorisée
Spécifique à un profil	Spécifique à un profil	Autorisée

Ext Channel Add Value

Cet attribut facultatif contient une valeur supplémentaire pour préciser davantage l'erreur de voie étendue. Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Si l'attribut "Ext Channel Error type" contient la valeur ACCUMULATIVE_INFO, la valeur de l'attribut doit avoir la signification selon le Tableau 139.

Autrement, la valeur est définie par le type d'erreur de voie ext.

Tableau 139 – Ext Channel Add Value pour Accumulative Info

Valeur	Signification
Bit0: 0x00	ChannelNumber n'est pas affecté.
Bit0: 0x01	ChannelNumber est affecté.
Bit1: 0x00	ChannelNumber +1 n'est pas affecté
Bit1: 0x01	ChannelNumber +1 est affecté
Bit2: 0x00	ChannelNumber +2 n'est pas affecté
Bit2: 0x01	ChannelNumber +2 est affecté
...	...
Bit30: 0x00	ChannelNumber +30 n'est pas affecté
Bit30: 0x01	ChannelNumber +30 est affecté
Bit31: 0x00	ChannelNumber +31 n'est pas affecté
Bit31: 0x01	ChannelNumber +31 est affecté

Type d'attribut: Unsigned32

Qualified Channel Qualifier

Cet attribut facultatif contient les données de diagnostic qualifiées. Il n'est présent que si l'attribut User Structure Identifier est mis sur QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Bit 0 et Bit 1 sont réservés. Bit 2 indique si Qualifier_2 est défini ou non, Bit 3 si Qualifier_3 est défini ou non, ... et Bit 31 si Qualifier_31 est défini ou non. Les valeurs doivent être utilisées en étant spécifiques à un profil.

Type d'attribut: Unsigned32

List of Manufacturer Specific Diagnosis Data

Cet attribut facultatif contient les informations de statut de diagnostic spécifique à un fabricant ou les informations de statut spécifique à un fabricant.

Cet attribut n'est présent que si l'attribut User Structure Identifier contient une valeur comprise dans la plage
MANUFACTURER_SPE_CIFIC_USI_0x0000 à
MANUFACTURER_SPE_CIFIC_USI_0x7FFF

Un élément de liste est constitué des attributs suivants:

Data

Cet attribut contient les données de diagnostic ou de statut spécifique à un fabricant.

Type d'attribut: L'un des types de données suivants doit être utilisé:

Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference, Floating Point, Network Time, VisibleString, Network Time Difference, OctetString

List of Nested Diagnosis Info

Cet attribut facultatif contient la liste des appareils subordonnés possédant des informations de diagnostic, de maintenance, qualifiées ou de diagnostic ou de statut spécifiques à un fabricant. L'attribut facultatif existe uniquement dans le cas où l'appareil indique le type d'erreur de voie NESTED_DIAGNOSIS_INDICATION.

Chaque élément de liste indique un appareil E/S situé dans le système E/S subordonné de cet appareil possédant des informations de diagnostic, de maintenance, qualifiées ou de diagnostic ou de statut spécifiques à un fabricant. Il est constitué des attributs suivants:

IP Address

Cet attribut doit être défini sur l'adresse IP de l'appareil E/S.

Type d'attribut: Unsigned32

Name Of Station Length

Cet attribut doit être défini sur la longueur du nom de station de l'appareil E/S.

Type d'attribut: Unsigned16

Name Of Station Value

Cet attribut doit être défini sur le nom de station de l'appareil E/S.

Type d'attribut: VisibleString[240]

7.3.4.2.3 Invocation de l'objet Diagnosis

Un appareil E/S doit invoquer un objet Diagnosis chaque Processus d'application qui doit contenir les informations de diagnostic de l'appareil E/S avec ses baies, sous-baies et voies défectueuses ou contenant des informations de maintenance.

7.3.4.3 Spécification du service Diagnosis**7.3.4.3.1 Add Diagnosis Entry**

Ce service est utilisé pour ajouter une entrée de diagnostic à l'ASE Diagnosis. Le Tableau 140 montre les paramètres du service.

Tableau 140 – Local Add Diagnosis Entry

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
API	M	
Slot Number	M	
Subslot Number	M	
List of Channel Diagnosis Data	M	
Channel Number	M	
Channel Properties	M	
User Structure Identifier	M	
Channel Error type	S	
Ext Channel Error type	S	
Ext Channel Add Value	S	
Qualified Channel Qualifier	S	
List of Manufacturer Specific Diagnosis Data	S	
Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

API

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Slot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Subslot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

List of Channel Diagnosis Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

User Structure Identifier

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Error type

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Error type

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Add Value

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Qualified Channel Qualifier

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur QUALIFIED_CHANNEL_DIAGNOSIS_USI.

List of Manufacturer Specific Diagnosis Data

Ce paramètre facultatif est constitué des éléments suivants.

Il est présent uniquement si l'attribut User Structure Identifier est mis de MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF.

Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Result(+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.4.3.2 Remove Diagnosis Entry

Ce service est utilisé pour supprimer une entrée de diagnostic à l'ASE Diagnosis. Le Tableau 141 montre les paramètres du service.

Tableau 141 – Local Remove Diagnosis Entry

Nom de paramètre	req	cnf
Argument	M	
AREP	M	
API	M	
Slot Number	M	
Subslot Number	M	
List of Channel Diagnosis Data	M	
Channel Number	M	
Channel Properties	M	
User Structure Identifier	M	
Channel Error type	S	
Ext Channel Error type	S	
Ext Channel Add Value	S	
Qualified Channel Qualifier	S	
List of Manufacturer Specific Diagnosis Data	S	
Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

API

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Slot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Subslot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

List of Channel Diagnosis Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

User Structure Identifier

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Error type

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Error type

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Add Value

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Qualified Channel Qualifier

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur QUALIFIED_CHANNEL_DIAGNOSIS_USI.

List of Manufacturer Specific Diagnosis Data

Ce paramètre facultatif est constitué des éléments suivants.

Il est présent uniquement si l'attribut User Structure Identifier est mis de MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF.

Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Result(+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

7.3.4.3.3 Local Diagnosis Event

Ce service est utilisé pour indiquer un événement de diagnostic détecté par des diagrammes protocolaires. Le Tableau 142 montre les paramètres du service.

En outre, les valeurs d'attributs de l'ASE Diagnosis doivent être mises en fonction du paramètre de service. De plus, un service Alarm doit être émis s'il est requis (par exemple pour le type d'alarme «Multicast Provider Communication Stopped» c'est-à-dire «arrêt de la communication fournisseur en multidiffusion»).

Tableau 142 – Local Diagnosis Event

Nom de paramètre	ind
Argument	M
API	M
Slot Number	M
Subslot Number	M
List of Channel Diagnosis Data	M
Channel Number	M
Channel Properties	M
User Structure Identifier	M
Channel Error type	S
Ext Channel Error type	S
Ext Channel Add Value	S
Qualified Channel Qualifier	S
List of Manufacturer Specific Diagnosis Data	S
Data	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à l'indication de service.

API

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Slot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Subslot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

List of Channel Diagnosis Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

User Structure Identifier

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Error type

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Error type

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Ext Channel Add Value

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI ou QUALIFIED_CHANNEL_DIAGNOSIS_USI.

Qualified Channel Qualifier

Ce paramètre facultatif contient la valeur de l'attribut correspondant de l'objet ASE.

Il n'est présent que si l'attribut User Structure Identifier est mis sur QUALIFIED_CHANNEL_DIAGNOSIS_USI.

List of Manufacturer Specific Diagnosis Data

Ce paramètre facultatif est constitué des éléments suivants.

Il est présent uniquement si l'attribut User Structure Identifier est mis de MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF.

Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

7.3.4.3.4 Définitions générales des données (Data)

Tous les services pour lire l'objet de diagnostic ont la même disposition de paramètre. Pour faciliter la description, paramètre est ainsi décrit une fois en amont des services. Les services eux-mêmes agissent comme un filtre pour les informations issues l'ASE Diagnosis.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1); pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique.

Le paramètre Data a une signification et une structure spécifiques au service. Le Tableau 143 montre les éléments spécifiques de Data pour ce service. Noter que selon le service particulier décrit ci-dessous, seules des parties des informations sont retournées (par exemple, aucune donnée de diagnostic spécifique à un fabricant...).

Tableau 143 – Définition générale de Data pour Diagnosis services

Nom de paramètre	éléments spécifiques à un service	rsp	cnf
Data (Données)	List of Diagnosis Data	M	M(=)
	API	M	M(=)
	Slot Number	M	M(=)
	Subslot Number	M	M(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	User Structure Identifier	M	M(=)
	List of Channel Diagnostic Data	S	S(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	Channel Error type	M	M(=)
	List of Ext Channel Diagnosis Data	S	S(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	Channel Error type	M	M(=)
	Ext Channel Error type	M	M(=)
	Ext Channel Add Value	M	M(=)
	List of Qualified Ext Channel Diagnosis Data	M	M(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	Channel Error type	M	M(=)
	Ext Channel Error type	M	M(=)
	Ext Channel Add Value	M	M(=)
	Qualified Channel Qualifier	M	M(=)
	Manufacturer Specific Diagnosis	S	S(=)
	Channel Number	M	M(=)
	Channel Properties	M	M(=)
	List of Data	M	M(=)
	Data (Données)	M	M(=)

Data

Ce paramètre contient les valeurs de l'objet qui a été lu.

List of Diagnosis Data

Ce paramètre contient une liste de données de diagnostic. Un élément de liste est constitué des paramètres suivants:

API

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Slot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Subslot Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Number

Ce paramètre doit être mis à 0x8000.

Channel Properties

Le champ ChannelProperties.Type, le champ ChannelProperties.Direction, le champ ChannelProperties.Maintenance doivent être mis à zéro.

Le champ ChannelProperties.Specifier doit être mis à APPEAR si au moins un ChannelProperties.Specifier dans les données de diagnostic est mis à APPEAR conjointement à ChannelProperties.Maintenance(=diagnosis). Sinon, le champ ChannelProperties.Spe_ci_fier doit être mis à DISAPPEAR.

User Structure Identifier

Ce paramètre indique le format des données autorisées.

Type d'attribut: Unsigned16

Valeurs permises: CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, QUALIFIED_CHANNEL_DIAGNOSIS_USI, MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF

NOTE L'appareil E/S détermine dans quel format un élément de diagnostic spécifique est retourné. Le demandeur peut déterminer si les données spécifiques à un fabricant doivent être retournées (en demandant «tous les codages») ou non (en demandant le «codage de voie»).

List of Channel Diagnostic Data

Ce paramètre facultatif contient une liste de données de diagnostic de voie liées à cette voie. Il ne doit être présent que si le paramètre User Structure Identifier est mis à CHANNEL_DIAGNOSIS_USI.

Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Error type

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

List of Ext Channel Diagnostic Data

Ce paramètre facultatif contient une liste de données de diagnostic de voie étendu liées à cette voie. Il ne doit être présent que si le paramètre User Structure Identifier est mis à EXT_CHANNEL_DIAGNOSIS_USI.

Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Error type

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Ext Channel Error type

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Ext Channel Add Value

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Qualified Ext Channel Diagnosis

Ce paramètre facultatif contient une liste de données de diagnostic de voie étendu liées à cette voie. Il ne doit être présent que si le paramètre User Structure Identifier est mis à EXT_CHANNEL_DIAGNOSIS_USI.

Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Error type

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Ext Channel Error type

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Ext Channel Add Value

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Qualified Channel Qualifier

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Manufacturer Specific Diagnosis

Ce paramètre facultatif contient une liste de données de diagnostic de voie étendu liées à cette voie. Il ne doit être présent que si le paramètre User Structure Identifier est mis sur MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF.

Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Channel Properties

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

List of Data

Ce paramètre contient une liste de données de diagnostic spécifique à un fabricant. Un élément de liste est constitué des paramètres suivants.

Data

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

7.3.4.3.5 Read Diagnosis in channel coding for one subslot

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans le codage de voie d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in channel coding for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant au sous-module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifié sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.6 Read Diagnosis in all codings for one subslot

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans tous les codages d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in all codings for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant au sous-module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis à DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.7 Read Maintenance required in channel coding for one subslot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in channel coding for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à la sous-baie adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifié sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.8 Read Maintenance required in all codings for one subslot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in all codings for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à la sous-baie adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.9 Read Maintenance demanded in channel coding for one subslot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in channel coding for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à la sous-baie adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données de diagnostic de voie, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.10 Read Maintenance demanded in all codings for one subslot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in all codings for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à la sous-baie adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.11 Read Diagnosis, Maintenance, Qualified and Status for one subslot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'une sous-baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis, Maintenance, Qualified and Status for one subslot appartient à un sous-module, qui est adressé par son numéro d'API, de baie et de sous-baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à la sous-baie adressée, et renvoie des informations pour ces voies, lorsque

- ChannelProperties.Maintenance est mis sur ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) et ChannelProperties.Specifier est mis sur APPEAR, ou
- ChannelProperties.Maintenance est mis sur DIAGNOSIS, ChannelProperties.Specifier est mis sur DISAPPEAR, et les données de diagnostic spécifiques au fabricant existent (à savoir les informations de statut).

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.12 Read Diagnosis in channel coding for one slot

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans le codage de voie d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in channel coding for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.13 Read Diagnosis in all codings for one slot

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans tous les codages d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in all codings for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.14 Read Maintenance required in channel coding for one slot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in channel coding for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.15 Read Maintenance required in all codings for one slot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in all codings for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.16 Read Maintenance demanded in channel coding for one slot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in channel coding for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules du sous-module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données de diagnostic de voies, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.17 Read Maintenance demanded in all codings for one slot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in all codings for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic de voie qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.18 Read Diagnosis, Maintenance, Qualified and Status for one slot

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis, Maintenance, Qualified and Status for one slot appartient à un module, qui est adressé par son numéro d'API et de baie. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque

- ChannelProperties.Maintenance est mis sur ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) et ChannelProperties.Specifier est mis sur APPEAR, ou
- ChannelProperties.Maintenance est mis sur DIAGNOSIS, ChannelProperties.Specifier est mis sur DISAPPEAR, et les données de diagnostic spécifiques au fabricant existent (à savoir les informations de statut).

Les données de diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic

spécifique à un fabricant est cependant toujours renvoyé dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.19 Read Diagnosis in channel coding for one AR

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans le codage de voie pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in channel coding for one AR appartient à une relation entre applications, qui est adressé par l'ARUUID ou s'il utilise une AR implicite par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules au sein de l'AR adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont renvoyées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.20 Read Diagnosis in all codings for one AR

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans tous les codages pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in all codings for one AR appartient à une relation entre applications, qui est adressé par l'ARUUID ou s'il utilise une AR implicite par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules au sein de l'AR adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Les données diagnostic de voie, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic de voie qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours renvoyé dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.21 Read Maintenance required in channel coding for one AR

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'une baie. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in channel coding for one AR appartient à une relation d'application, qui est adressé par l'ARUUID ou, si une AR implicite est utilisée, par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'AR adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.22 Read Maintenance required in all codings for one AR

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in all codings for one AR appartient à une relation d'application, qui est adressée par l'ARUUID ou, si une AR implicite est utilisée, par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'AR adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifiées et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.23 Read Maintenance demanded in channel coding for one AR

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in channel coding for one AR appartient à une relation d'application, qui est adressé par l'ARUUID ou, si une AR implicite est utilisée, par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'AR adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.24 Read Maintenance demanded in all codings for one AR

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in all codings for one AR appartient à une relation d'application, qui est adressée par l'ARUUID ou, si une AR implicite est utilisée, par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules de l'AR adressée, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.25 Read Diagnosis, Maintenance, Qualified and Status for one AR

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis, Maintenance, Qualified and Status for one AR appartient à une relation entre applications, qui est adressé par l'ARUUID ou s'il utilise une AR implicite par Target AR UUID. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules de l'AR adressée, et renvoie des informations pour ces voies, lorsque

- ChannelProperties.Maintenance est mis sur ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) et ChannelProperties.Specifier est mis sur APPEAR, ou
- ChannelProperties.Maintenance est mis sur DIAGNOSIS, ChannelProperties.Specifier est mis sur DISAPPEAR, et les données de diagnostic spécifiques au fabricant existent (à savoir les informations de statut).

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.26 Read Diagnosis in channel coding for one API

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans le codage de voie pour une AR. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in channel coding for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'API adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas renvoyées

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.27 Read Diagnosis in all codings for one API

Ce service confirmé peut être utilisé pour lire les données de diagnostic dans tous les codages pour un API. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis in all codings for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à l'API adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur DIAGNOSIS et ChannelProperties.Specifier est mis sur APPEAR.

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.28 Read Maintenance required in channel coding for one API

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'un API. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in channel coding for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'API adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis à MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis à APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas renvoyées

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.29 Read Maintenance required in all codings for one API

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'un API. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance required in all codings for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'API adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_REQUIRED et ChannelProperties.Specifier est mis sur APPEAR.

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.30 Read Maintenance demanded in channel coding for one API

Ce service confirmé peut être utilisé pour lire les données de maintenance dans le codage de voie d'un API. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in channel coding for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant à tous les sous-modules de l'API adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Seules les données des voies de diagnostic, les données de diagnostic de voie étendu et les données de diagnostic qualifiées sont retournées, ou, autrement dit, les données de diagnostic spécifiques à un fabricant ne sont pas renvoyées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné.

7.3.4.3.31 Read Maintenance demanded in all codings for one API

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'un API. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Maintenance demanded in all codings for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules de l'API adressé, et renvoie des informations pour ces voies, lorsque ChannelProperties.Maintenance est mis sur MAINTENANCE_DEMANDED et ChannelProperties.Specifier est mis sur APPEAR.

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.32 Read Diagnosis, Maintenance, Qualified and Status for one API

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'un API. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis, Maintenance, Qualified and Status for one API appartient à un AP, qui est adressé par son API. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules du module adressé, et renvoie des informations pour ces voies, lorsque

- ChannelProperties.Maintenance est mis sur ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) et ChannelProperties.Specifier est mis sur APPEAR, ou
- ChannelProperties.Maintenance est mis sur DIAGNOSIS, ChannelProperties.Specifier est mis sur DISAPPEAR, et les données de diagnostic spécifiques au fabricant existent (à savoir les informations de statut).

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours renvoyé dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.33 Read Diagnosis, Maintenance, Qualified and Status for one device

Ce service confirmé peut être utilisé pour lire les données de maintenance dans tous les codages d'un appareil. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1). Pour la hiérarchie des codes d'erreur de l'objet Record Data, le Paragraphe 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques au service. Voir 7.3.4.3.4 pour sa description générale.

Le service Diagnosis, Maintenance, Qualified and Status for one device appartient à l'appareil. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Ce service filtre l'objet de diagnostic concernant les voies appartenant aux sous-modules de l'appareil adressé, et renvoie des informations pour ces voies, lorsque

- ChannelProperties.Maintenance est mis sur ANY (DIAGNOSIS, MAINTENANCE_REQUIRED, MAINTENANCE_DEMANDED, QUALIFIED_DIAGNOSIS) et ChannelProperties.Specifier est mis sur APPEAR, ou
- ChannelProperties.Maintenance est mis sur DIAGNOSIS, ChannelProperties.Specifier est mis sur DISAPPEAR, et les données de diagnostic spécifiques au fabricant existent (à savoir les informations de statut).

Les données des voies de diagnostic, les données de diagnostic de voie étendu, les données de diagnostic qualifié et les données de diagnostic spécifiques à un fabricant sont retournées.

NOTE L'appareil E/S détermine dans quel format (données de diagnostic de voie, données de diagnostic de voie étendu ou données de diagnostic qualifiées) un élément de diagnostic spécifique est retourné. Le diagnostic spécifique à un fabricant est cependant toujours retourné dans le format des données de diagnostic spécifiques à un fabricant.

7.3.4.3.34 Read Nested Diagnosis Information

Ce service facultatif peut être utilisé pour lire les informations de diagnostic imbriquées. Ce service doit être utilisé conjointement à l'AR implicite (Implicit AR), à l'IO AR ou à la Supervisor AR.

La structure et la signification des paramètres sont les mêmes que celles décrites dans le service de lecture (voir 7.3.1.4.1.1); Pour la hiérarchie des codes d'erreur de l'objet Record Data, 7.3.1.3.5 s'applique. Le paramètre Data a une signification et une structure spécifiques à un service. Le Tableau 144 indique les éléments spécifiques de Data pour ce service.

Le service Nested Diagnosis Information appartient à l'appareil. Le paramètre Index identifie l'indice d'enregistrement de données tel que spécifié dans la CEI 61158-6-10.

Tableau 144 – Eléments Data de Read Nested Diagnosis Information

Nom de paramètre	éléments spécifiques à un service	rsp	cnf
Data	Number of Entries	M	M(=)
	List of Nested Diagnosis Info	M	M(=)
	Slot Number	M	M(=)
	IP Address	M	M(=)
	Station Name Length	M	M(=)
	Name Of Station Value	M	M(=)

Data

Ce paramètre contient les valeurs de l'objet qui a été lu.

Number Of Entries

Ce paramètre indique le nombre d'entrées dans la liste suivante.

Type d'attribut: Unsigned16

List of Nested Diagnosis Info

Ce paramètre est constitué des éléments de liste suivants:

IP Address

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Station Name Length

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

Name Of Station Value

Ce paramètre contient la valeur de l'attribut correspondant de l'objet ASE.

7.3.4.4 Comportement des objets Diagnosis**7.3.4.4.1 Comportement général de l'objet Diagnosis**

Une entrée d'un objet Diagnosis doit maintenir différents états pour "diagnosis" (diagnostic), "maintenance required" (maintenance requise), "maintenance demanded" (maintenance exigée), et "qualified_2 à qualified_32 (qualifiée_ 2 à _32).

7.3.4.4.2 Comportement d'entrée de diagnostic**7.3.4.4.2.1 Définitions des primitives****7.3.4.4.2.1.1 Primitives échangées entre diagrammes distants**

Les primitives de service, y compris leurs paramètres associés, qui sont émises ou reçues par le diagramme d'états d'entrée de diagnostic (DIAG_DIAG) sont décrites dans la définition du service et indiquées dans le Tableau 145.

Tableau 145 – Primitives distantes émises ou reçues par DIAG_DIAG

Primitive	Source	Destination	Paramètres associés	Fonctions
Alarm.req	DIAG_DIAG	FAL	voir 7.3.5.3	—

7.3.4.4.2.1.2 Primitives échangées entre diagrammes locaux

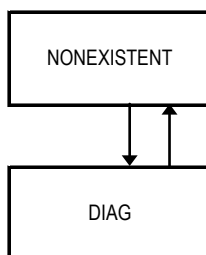
Les primitives de service locales, y compris leurs paramètres associés, qui sont émises ou reçues par DIAG_DIAG sont décrites dans la définition du service et indiquées dans le Tableau 146.

Tableau 146 – Primitives locales émises ou reçues par DIAG_DIAG

Primitive	Source	Destination	Paramètres associés	Fonctions
AddDiagnosisEntry_req ()	FAL	DIAG_DIAG	voir spécification de service	—
AddDiagnosisEntry_cnf ()	DIAG_DIAG	FAL	voir spécification de service	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_DIAG	voir spécification de service	—
RemoveDiagnosisEntry_cnf ()	DIAG_DIAG	FAL	voir spécification de service	—

7.3.4.4.2.2 Diagramme de transitions d'états

Le diagramme des transitions d'états est montré à la Figure 28.



Légende

Anglais	Français
NONEXISTENT	NONEXISTENT (Non existant)
DIAG	DIAG (Diagnostic)

Figure 28 – Diagramme de transition d'états DIAG_DIAG

Etats du DIAG_DIAG

NONEXISTENT

Cet état indique qu'aucune entrée de diagnostic n'existe ou qu'une entrée existante est supprimée et donc qu'une demande d'alarme est envoyée.

DIAG

Cet état indique qu'une entrée de diagnostic existe et qu'une demande d'alarme est envoyée.

7.3.4.4.2.3 Description de diagramme d'états

Ce diagramme est destiné à montrer pour l'utilisateur que le comportement du diagnostic s'articule en deux étapes. La première étape est utilisée pour enregistrer les données de diagnostic vers l'ASE Diagnosis (diagnostic) alors que la seconde étape envoie une demande d'alarme avec les données de diagnostic.

Si le diagnostic disparaît, deux étapes sont également nécessaires. Les données locales doivent être mises à jour et l'appareil de commande doit être informé lors d'une demande d'alarme.

Afin de simplifier le diagramme d'états, la confirmation n'est pas incluse.

7.3.4.4.2.4 Table d'états

Le Tableau 147 définit la table d'états DIAG_DIAG.

Tableau 147 – Table d'états DIAG_DIAG

#	Etat courant	Evénement / Condition => Action	Prochain état
1	NONEXISTENT	AddDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType= (CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => CreateDiagnosisEntry () UpdateAPDUStatus (For all related ARs) Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	DIAG
2	DIAG	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => RemoveDiagnosisEntry () UpdateAPDUStatus (For all related ARs) Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.2.5 Fonctions, macros, temporisateurs et variables

Le Tableau 148 contient les fonctions, les macros, les temporisateurs et les variables utilisés par DIAG_DIAG ainsi que leurs arguments et leurs descriptions.

Tableau 148 – Fonctions, macros, temporisateurs et variables utilisés par DIAG_DIAG

Nom	Type	Fonction/Signification
DeleteDiagnosisEntry	Fonction	Cette fonction locale supprimer l'entrée de diagnostic du format donné pour la notification d'alarme (Channel Properties.Specifier:= APPEAR) en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data

Nom	Type	Fonction/Signification
BuildDiagnosisAlarmDisappears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme, l'entrée de diagnostic (Channel Properties. Specifier) doit contenir les informations récapitulatives de diagnostic pour l'ensemble API/module/sous-module/voie pour sélectionner DISAPPEAR ou DISAPPEAR_OTHER_EXIST en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Fonction	Cette fonction locale retourne l'entrée de spécificateur d'alarme pour la notification d'alarme, le spécificateur d'alarme doit contenir les informations d'alarme récapitulées pour l'ensemble API/module/sous-module (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) et pour tous les API/modules/sous-module des AR connexes (AR Diagnosis State) en utilisant les paramètres suivants: API, Slot, Subslot
BuildSummarizedMaintenanceStatus	Fonction	Cette fonction locale renvoie l'entrée Maintenance Status pour la notification d'alarme, elle doit contenir les informations récapitulatives de maintenance des API/module/sous-module (requis, exigée ou qualifiée 2 à 31) si aucune maintenance n'est présente, rien ne doit être renvoyé et le paramètre de service facultatif doit être omis dans la notification d'alarme en utilisant les paramètres suivants: API, Slot, Subslot
CreateDiagnosisEntry	Fonction	Cette fonction locale invoque une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Fonction	Cette fonction locale renvoie la priorité d'alarme pour le type d'alarme dédié à une certaine AR en utilisant le paramètre suivant: AREP
RemoveDiagnosisEntry	Fonction	Cette fonction locale retire une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties

Nom	Type	Fonction/Signification
ApplicationDiagnosisDetected	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer un diagnostic en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer qu'un diagnostic n'existe plus en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType
UpdateAPDUStatus	Macro	Cette macro met le fanion "Problem indicator Flag" à TRUE pour toutes les AR connexes si au moins une entrée de diagnostic "apparaît" pour l'API/module/sous-module existe; autrement, il doit être mis à FALSE

7.3.4.4.3 Comportement d'entrée de maintenance requise

7.3.4.4.3.1 Définitions des primitives

7.3.4.4.3.1.1 Primitives échangées entre diagrammes distants

Les primitives de service, y compris leurs paramètres associés, qui sont émises ou reçues par le diagramme d'états d'entrée de maintenance requise (DIAG_MR) sont décrites dans la définition du service et indiquées dans le Tableau 149.

Tableau 149 – Primitives distantes émises ou reçues par DIAG_MR

Primitive	Source	Destination	Paramètres associés	Fonctions
Alarm.req	DIAG_MR	FAL	voir 7.3.5.3	—

7.3.4.4.3.1.2 Primitives échangées entre diagrammes locaux

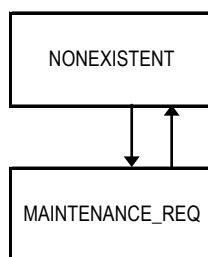
Les primitives de service locales, y compris leurs paramètres associés, qui sont émises ou reçues par DIAG_MR sont décrites dans la définition du service et indiquées dans le Tableau 150.

Tableau 150 – Primitives locales émises ou reçues par DIAG_MR

Primitive	Source	Destination	Paramètres associés	Fonctions
AddDiagnosisEntry_req ()	FAL	DIAG_MR	voir spécification de service	—
AddDiagnosisEntry_cnf ()	DIAG_MR	FAL	voir spécification de service	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_MR	voir spécification de service	—
RemoveDiagnosisEntry_cnf ()	DIAG_MR	FAL	voir spécification de service	—

7.3.4.4.3.2 Diagramme de transitions d'états

Le diagramme des transitions d'états est montré à la Figure 29.



Légende

Anglais	Français
NONEXISTENT	NONEXISTENT (Non existant)
MAINTENANCE_DEM	MAINTENANCE_DEM (Maintenance exigée)

Figure 29 – Diagramme de transition d'états DIAG_MR

Etats du DIAG_MR

NONEXISTENT

Cet état indique qu'aucune entrée de diagnostic n'existe ou qu'une entrée existante est supprimée et donc qu'une demande d'alarme est envoyée

MAINTENANCE_REQ

Cet état indique qu'une entrée de diagnostic existe et qu'une demande d'alarme est envoyée

7.3.4.4.3.3 Description de diagramme d'états

Ce diagramme est destiné à montrer pour l'utilisateur que le comportement du diagnostic s'articule en deux étapes. La première étape est utilisée pour enregistrer les données de diagnostic maintenance required (maintenance requise) vers l'ASE de diagnostic alors que la seconde étape envoie une demande d'alarme avec les données de diagnostic.

Si le diagnostic disparaît, deux étapes sont également nécessaires. Les données locales doivent être mises à jour et l'appareil de commande doit être informé lors d'une demande d'alarme.

Afin de simplifier le diagramme d'états, la confirmation n'est pas incluse.

7.3.4.4.3.4 Table d'états

Le Tableau 151 définit la table d'états DIAG_MR.

Tableau 151 – Table d'états DIAG_MR

#	Etat courant	Événement / Condition => Action	Prochain état
1	NONEXISTENT	AddDiagnosisEntry_req () /ChannelProperties.Maintenance Required=TRUE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType= (CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => CreateDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	MAINTENANCE_REQ
2	MAINTENANCE_REQ	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=TRUE && ChannelProperties.Maintenance Demanded=FALSE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => DeleteDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.3.5 Fonctions, macros, temporisateurs et variables

Le Tableau 152 contient les fonctions, les macros, les temporisateurs et les variables utilisés par DIAG_MR ainsi que leurs arguments et leurs descriptions.

Tableau 152 – Fonctions, macros, temporisateurs et variables utilisés par DIAG_MR

Nom	Type	Fonction/Signification
BuildDiagnosisAlarmAppears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme (Channel Properties. Specifier:= APPEAR) en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data

Nom	Type	Fonction/Signification
BuildDiagnosisAlarmDisappears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme, l'entrée de diagnostic (Channel Properties. Specifier) doit contenir les informations récapitulatives de diagnostic pour l'ensemble API/module/sous-module/voie pour sélectionner DISAPPEAR ou DISAPPEAR_OTHER_EXIST en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Fonction	Cette fonction locale retourne l'entrée de spécificateur d'alarme pour la notification d'alarme, le spécificateur d'alarme doit contenir les informations d'alarme récapitulées pour l'ensemble API/module/sous-module (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) et pour tous les API/modules/sous-module des AR connexes (AR Diagnosis State) en utilisant les paramètres suivants: API, Slot, Subslot
BuildSummarizedMaintenanceStatus	Fonction	Cette fonction locale renvoie l'entrée Maintenance Status pour la notification d'alarme, elle doit contenir les informations récapitulatives de maintenance des API/module/sous-module (requise, exigée ou qualifiée 2 à 31) si aucune maintenance n'est présente, rien ne doit être renvoyé et le paramètre de service facultatif doit être omis dans la notification d'alarme en utilisant les paramètres suivants: API, Slot, Subslot
CreateDiagnosisEntry	Fonction	Cette fonction locale invoque une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Fonction	Cette fonction locale renvoie la priorité d'alarme pour le type d'alarme dédié à une certaine AR en utilisant le paramètre suivant: AREP
DeleteDiagnosisEntry	Fonction	Cette fonction locale supprime une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties

Nom	Type	Fonction/Signification
ApplicationDiagnosisDetected	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer un diagnostic en utilisant les paramètres suivants. API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer qu'un diagnostic n'existe plus en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType

7.3.4.4.4 Comportement d'entrée de maintenance exigée

7.3.4.4.4.1 Définitions des primitives

7.3.4.4.4.1.1 Primitives échangées entre diagrammes distants

Les primitives de service, y compris leurs paramètres associés, qui sont émises ou reçues par le diagramme d'états d'entrée de maintenance exigée (DIAG_MD) sont décrites dans la définition du service et indiquées dans le Tableau 153.

Tableau 153 – Primitives distantes émises ou reçues par DIAG_MD

Primitive	Source	Destination	Paramètres associés	Fonctions
Alarm.req	DIAG_MD	FAL	voir 7.3.5.3	—

7.3.4.4.4.1.2 Primitives échangées entre diagrammes locaux

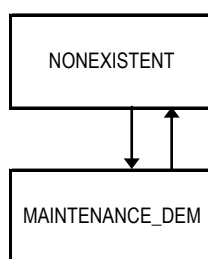
Les primitives de service locales, y compris leurs paramètres associés, qui sont émises ou reçues par DIAG_MD sont décrites dans la définition du service et indiquées dans le Tableau 154.

Tableau 154 – Primitives locales émises ou reçues par DIAG_MD

Primitive	Source	Destination	Paramètres associés	Fonctions
AddDiagnosisEntry_req ()	FAL	DIAG_MD	voir spécification de service	—
AddDiagnosisEntry_cnf ()	DIAG_MD	FAL	voir spécification de service	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_MD	voir spécification de service	—
RemoveDiagnosisEntry_cnf ()	DIAG_MD	FAL	voir spécification de service	—

7.3.4.4.4.2 Diagramme de transitions d'états

Le diagramme des transitions d'états est montré à la Figure 30.



Légende

Anglais	Français
NONEXISTENT	NONEXISTENT (Non existant)
MAINTENANCE_DEM	MAINTENANCE_DEM (Maintenance exigée)

Figure 30 – Diagramme de transition d'états DIAG_MD

Etats du DIAG_MD

NONEXISTENT

Cet état indique qu'aucune entrée de diagnostic n'existe ou qu'une entrée existante est supprimée et donc qu'une demande d'alarme est envoyée

MAINTENANCE_DEM

Cet état indique qu'une entrée de diagnostic existe et qu'une demande d'alarme est envoyée

7.3.4.4.4.3 Description de diagramme d'états

Ce diagramme est destiné à montrer pour l'utilisateur que le comportement du diagnostic s'articule en deux étapes. La première étape est utilisée pour enregistrer les données de diagnostic "maintenance demanded" (maintenance exigée) auprès de l'ASE de diagnostic alors que la seconde étape envoie une demande d'alarme avec les données de diagnostic.

Si le diagnostic disparaît, deux étapes sont également nécessaires. Les données locales doivent être mises à jour et l'appareil de commande doit être informé lors d'une demande d'alarme.

Afin de simplifier le diagramme d'états, la confirmation n'est pas incluse.

7.3.4.4.4.4 Table d'états

Le Tableau 155 définit la table d'états DIAG_MD.

Tableau 155 – Table d'états DIAG_MD

#	Etat courant	Evénement / Condition => Action	Prochain état
1	NONEXISTENT	AddDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType= (CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => CreateDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	MAINTENANCE_DEM
2	MAINTENANCE_DEM	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => DeleteDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.4.5 Fonctions, macros, temporisateurs et variables

Le Tableau 156 contient les fonctions, les macros, les temporisateurs et les variables utilisés par DIAG_MD ainsi que leurs arguments et leurs descriptions.

**Tableau 156 – Fonctions, macros, temporisateurs et variables utilisés
par l'entrée de maintenance exigée**

Nom	Type	Fonction/Signification
BuildDiagnosisAlarmAppears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme (Channel Properties. Specifier:= APPEAR) en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data

Nom	Type	Fonction/Signification
BuildDiagnosisAlarmDisappears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme, l'entrée de diagnostic (Channel Properties. Specifier) doit contenir les informations récapitulatives de diagnostic pour l'ensemble API/module/sous-module/voie pour sélectionner DISAPPEAR ou DISAPPEAR_OTHER_EXIST en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Fonction	Cette fonction locale retourne l'entrée de spécificateur d'alarme pour la notification d'alarme, le spécificateur d'alarme doit contenir les informations d'alarme récapitulées pour l'ensemble API/module/sous-module (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) et pour tous les API/modules/sous-module des AR connexes (AR Diagnosis State) en utilisant les paramètres suivants: API, Slot, Subslot
BuildSummarizedMaintenanceStatus	Fonction	Cette fonction locale renvoie l'entrée Maintenance Status pour la notification d'alarme, elle doit contenir les informations récapitulatives de maintenance des API/module/sous-module (requis, exigée ou qualifiée 2 à 31) si aucune maintenance n'est présente, rien ne doit être renvoyé et le paramètre de service facultatif doit être omis dans la notification d'alarme en utilisant les paramètres suivants: API, Slot, Subslot
CreateDiagnosisEntry	Fonction	Cette fonction locale invoque une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Fonction	Cette fonction locale renvoie la priorité d'alarme pour le type d'alarme dédié à une certaine AR en utilisant le paramètre suivant: AREP
DeleteDiagnosisEntry	Fonction	Cette fonction locale supprime une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties

Nom	Type	Fonction/Signification
ApplicationDiagnosisDetected	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer un diagnostic en utilisant les paramètres suivants. API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer qu'un diagnostic n'existe plus en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType

7.3.4.4.5 Comportement d'entrée "qualified" (qualifiée)

7.3.4.4.5.1 Définitions des primitives

7.3.4.4.5.1.1 Primitives échangées entre diagrammes distants

Les primitives de service, y compris leurs paramètres associés, qui sont émises ou reçues par le diagramme d'états d'entrée qualifiée (DIAG_QUALIFIED) sont décrites dans la définition du service et indiquées dans le Tableau 157.

Tableau 157 – Primitives distantes émises ou reçues par DIAG_QUALIFIED

Primitive	Source	Destination	Paramètres associés	Fonctions
Alarm.req	DIAG_QUALIFIED	FAL	voir 7.3.5.3	—

7.3.4.4.5.1.2 Primitives échangées entre diagrammes locaux

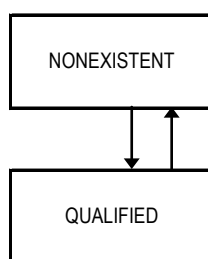
Les primitives de service locales, y compris leurs paramètres associés, qui sont émises ou reçues par DIAG_QUALIFIED sont décrites dans la définition du service et indiquées dans le Tableau 158.

Tableau 158 – Primitives locales émises ou reçues par DIAG_QUALIFIED

Primitive	Source	Destination	Paramètres associés	Fonctions
AddDiagnosisEntry_req ()	FAL	DIAG_QUALIFIED	voir spécification de service	—
AddDiagnosisEntry_cnf ()	DIAG_QUALIFIED	FAL	voir spécification de service	—
RemoveDiagnosisEntry_req ()	FAL	DIAG_QUALIFIED	voir spécification de service	—
RemoveDiagnosisEntry_cnf ()	DIAG_QUALIFIED	FAL	voir spécification de service	—

7.3.4.4.5.2 Diagramme de transitions d'états

Le diagramme des transitions d'états est montré à la Figure 31.



Légende

Anglais	Français
NONEXISTENT	NONEXISTENT (Non existant)
QUALIFIED	QUALIFIED (Qualifié)

Figure 31 – Diagramme de transition d'états DIAG_QUALIFIED

Etats du DIAG_QUALIFIED

NONEXISTENT

Cet état indique qu'aucune entrée de diagnostic n'existe ou qu'une entrée existante est supprimée et donc qu'une demande d'alarme est envoyée

QUALIFIED

Cet état indique qu'une entrée de diagnostic existe et qu'une demande d'alarme est envoyée

7.3.4.4.5.3 Description de diagramme d'états

Ce diagramme est destiné à montrer pour l'utilisateur que le comportement du diagnostic s'articule en deux étapes. La première étape est utilisée pour enregistrer les données de diagnostic vers l'ASE de diagnostic alors que la seconde étape envoie une demande d'alarme avec les données de diagnostic.

Si le diagnostic disparaît, deux étapes sont également nécessaires. Les données locales doivent être mises à jour et l'appareil de commande doit être informé lors d'une demande d'alarme.

Afin de simplifier le diagramme d'états, la confirmation n'est pas incluse.

7.3.4.4.5.4 Table d'états

Le Tableau 159 définit la table d'états qui existe pour chaque entrée qualifiée (2 à 31).

Tableau 159 – Table d'états DIAG_QUALIFIED

#	Etat courant	Événement / Condition => Action	Prochain état
1	NONEXISTENT	AddDiagnosisEntry_req (Data=NIL) /ChannelProperties.Maintenance Required=TRUE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType= (QUALIFIED_EXT_CHANNELDIAGNOSIS) => CreateDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmAppears (Data=NIL) Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	QUALIFIED

#	Etat courant	Evénement / Condition => Action	Prochain état
2	QUALIFIED	RemoveDiagnosisEntry_req () /ChannelProperties.Maintenance Required=FALSE && ChannelProperties.Maintenance Demanded=TRUE && DiagnosisType=(CHANNELDIAGNOSIS EXT_CHANNELDIAGNOSIS QUALIFIED_EXT_CHANNELDIAGNOSIS MANUFACTURER_SPECIFICDIAGNOSIS) => DeleteDiagnosisEntry () Alarm Item.Diagnosis = BuildDiagnosisAlarmDisappears () Alarm Specifier=BuildSummarizedAlarmSpecifier () Alarm Item. Maintenance Status=BuildSummarizedMaintenanceStatus () Alarm Priority=GetAREPAlarmPriority () Alarm.req ()	NONEXISTENT

7.3.4.4.5.5 Macros

Le Tableau 160 contient les fonctions, les macros, les temporisateurs et les variables utilisés par DIAG_QUALIFIED ainsi que leurs arguments et leurs descriptions.

Tableau 160 – Fonctions, macros, temporisateurs et variables utilisés par DIAG_QUALIFIED

Nom	Type	Fonction/Signification
BuildDiagnosisAlarmAppears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme (Channel Properties. Specifier:= APPEAR) en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
BuildDiagnosisAlarmDisappears	Fonction	Cette fonction locale retourne l'entrée de diagnostic du format donné pour la notification d'alarme, l'entrée de diagnostic (Channel Properties. Specifier) doit contenir les informations récapitulatives de diagnostic pour l'ensemble API/module/sous-module/voie pour sélectionner DISAPPEAR ou DISAPPEAR_OTHER_EXIST en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, Alarm Item Format, ChannelErrorType, ExtChannelErrorType
BuildSummarizedAlarmSpecifier	Fonction	Cette fonction locale retourne l'entrée de spécificateur d'alarme pour la notification d'alarme, le spécificateur d'alarme doit contenir les informations d'alarme récapitulées pour l'ensemble API/module/sous-module (Submodule Diagnosis State, Submodule Channel Diagnosis, Submodule Manufacturer Specific Diagnosis) et pour tous les API/modules/sous-module des AR connexes (AR Diagnosis State) en utilisant les paramètres suivants: API, Slot, Subslot

Nom	Type	Fonction/Signification
BuildSummarizedMaintenanceStatus	Fonction	Cette fonction locale renvoie l'entrée Maintenance Status pour la notification d'alarme, elle doit contenir les informations récapitulatives de maintenance des API/module/sous-module (requis, exigée ou qualifiée 2 à 31) si aucune maintenance n'est présente, rien ne doit être renvoyé et le paramètre de service facultatif doit être omis dans la notification d'alarme en utilisant les paramètres suivants: API, Slot, Subslot
CreateDiagnosisEntry	Fonction	Cette fonction locale invoque une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
GetAREPAlarmPriority	Fonction	Cette fonction locale renvoie la priorité d'alarme pour le type d'alarme dédié à une certaine AR en utilisant le paramètre suivant: AREP
DeleteDiagnosisEntry	Fonction	Cette fonction locale supprime une entrée de diagnostic du type donné dans l'objet Diagnosis en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties
ApplicationDiagnosisDetected	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer un diagnostic en utilisant les paramètres suivants. API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ChannelProperties, ExtChannelErrorType, AddInfo, QualifiedChannelQualifier, Data
ApplicationDiagnosisRemoved	Macro	Cet événement est appelé par le processus d'application d'un appareil E/S pour indiquer qu'un diagnostic n'existe plus en utilisant les paramètres suivants: API, Slot, Subslot, ChannelNumber, DiagnosisType, ChannelErrorType, ExtChannelErrorType, QualifiedChannelQualifier

7.3.4.4.6 Optimisation pour objets Diagnosis

7.3.4.4.6.1 Caractéristique commune

Un objet de diagnostic doit être complètement adressé par API, Slot Number, Subslot Number, Channel Number, Channel Error type, Ext Channel Error type et Channel Properties (en utilisant uniquement Maintenance required, Maintenance demanded et Specifier).

7.3.4.4.6.2 Niveau Subslot Number

Si tous les objets de diagnostic du niveau d'adresse sous-baie (un Subslot Number) sont supprimés en une seule fois, une alarme "Diagnosis disappears" sans un élément d'alarme doit être générée.

Cela signale que l'ensemble des diagnostics, maintenances (requis et exigés) et données qualifiées ultérieures a disparu

7.3.4.4.6.3 Niveau Channel Error type

Si tous les objets de diagnostic du niveau d'adresse Channel Error type (un Channel Error type) sont supprimés en une seule fois, un élément d'alarme avec le contenu suivant doit être généré:

- Channel Number
- Channel Properties
 - Type
 - Direction
 - Maintenance Required et Maintenance Demanded doivent être zéro
 - Spécifier doit être zéro
- Channel Error type

Cela signale que l'ensemble des diagnostics, maintenances (requis et exigés) et données qualifiées ultérieures a disparu

NOTE Ultérieurs dans ce cas signifie que tous les objets de diagnostic adressés par Ext Channel Error Types ont également disparu.

7.3.5 ASE Alarm

7.3.5.1 Vue d'ensemble

Le modèle d'alarme permet le transfert d'une alarme de l'appareil E/S à l'appareil de commande E/S assigné ou vice versa et l'acquiescement explicite de l'alarme.

Pour l'émission d'une alarme, les conditions suivantes doivent être remplies:

- l'IO AR doit être établie
- le sous-module source doit être possédé par l'IO AR
- le sous-module source doit être existant (aucune baie vide et aucun sous-module erroné).

Une alarme doit toujours être émise; le puits doit répondre par un acquiescement négatif "not supported" («non pris en charge») si aucun descripteur pour le type Alarm n'est installé.

Le Tableau 161 donne un aperçu des types d'alarmes définies, leur utilisation et si la prise en charge de l'alarme est obligatoire ou facultative.

Les types d'alarmes indiqués comme «attachées à l'ASE Diagnosis» sont émis, chaque fois que des entrées sont ajoutées ou retirées dans l'ASE Diagnosis. Ces alarmes permettent ainsi le maintien de l'état de diagnostic réel d'un appareil E/S. Les services de l'ASE de diagnostic permettent obtenir l'état actuel du diagnostic.

Notez que les alarmes liées à l'ASE diagnostic peuvent acheminer plusieurs éléments de diagnostic pour une voie (pour réduire le nombre d'alarmes à envoyer)

Les colonnes «charge utile d'alarme» définissent les éléments qui font partie de la charge utile d'alarme d'une alarme particulière.

Tableau 161 – Alarm type

Alarm type	Usage	Utilisation	Attaché à l'ASE Diagnosis	Alarm Payload				
				Maintenance Item	Diagnosis Item	Upload & Retrieval Parameter Item	Manufacturer Alarm Item	Profile Alarm Item
Diagnosis (Diagnostic)	La valeur Diagnosis doit être utilisée pour indiquer des éléments qui apparaissent ou disparaissent pour une ou plusieurs voies.	Obligatoire	Oui	a	b	—	—	—
Process	Indique une alarme provenant du processus commandé (usage spécifique à une application et à un processus)	Facultatif	Non	—	—	—	c	—
Pull	Doit indiquer une alarme si un module/sous-module utilisé est "pulled" (tiré).	Obligatoire	Non	—	—	—	c	—
Plug	Doit indiquer une alarme si un sous-module demandé est "plugged" ("branché").	Obligatoire	Non	—	—	—	c	—
Status	Indique un changement de statut au sein d'un sous-module.	Facultatif	Non	—	—	—	c	—
Update	Indique un changement de paramètre pour un sous-module.	Facultatif	Non	—	—	—	c	—
Media Redundancy	Indique un changement du statut de redondance de supports.	Obligatoire pour la prise en charge de la redondance des supports	Oui	a	b	—	—	—
Controlled by supervisor	Doit indiquer une alarme si un sous-module utilisé a été verrouillé par un superviseur E/S. "Pull" logique d'un sous-module pour retirer le droit de propriété	Obligatoire	Non	—	—	—	c	—
Released	Doit indiquer une alarme si un sous-module demandé a été déverrouillé par un superviseur E/S, un appareil de commande E/S ou par des moyens locaux de d'appareil E/S. "plug" logique d'un sous-module pour renvoyer la propriété ou déclencher un reparamétrage	Obligatoire	Non	—	—	—	c	—
Plug Wrong Submodule	Doit indiquer une alarme si un sous-module erroné pour un module/sous-module demandé est branché.	Obligatoire	Non	—	—	—	c	—
Return Of Submodule	Doit indiquer un changement de l'IOCS/IOPS de BAD à GOOD pour un sous-module	Obligatoire	Non	—	—	—	c	—

Alarm type	Usage	Utilisation	Attaché à l'ASE Diagnosis	Alarm Payload				
				Maintenance Item	Diagnosis Item	Upload & Retrieval Parameter Item	Manufacturer Alarm Item	Profile Alarm Item
Diagnosis Disappears	La valeur "Diagnosis Disappears" doit être utilisé seulement - pour indiquer toute disparition sans un élément d'alarme Alarm Item (optimisation) - pour indiquer des éléments disparaissants détaillés.	Obligatoire	Oui	—	—	—	d	—
Multicast Communication Mismatch Notification	Un consommateur en multidiffusion doit indiquer que la CR de multidiffusion a eu un problème.	Obligatoire pour la prise en charge de la multidiffusion	Oui	a	b	—	—	—
Port Data Change Notification	Indique un changement de données de port, par exemple: liaison up (établie) ou down (perdue).	Obligatoire	Oui	a	b	—	—	—
Sync Data Changed Notification	Indique un changement de synchronisation d'horloge.	Obligatoire pour la prise en charge de la synchronisation d'horloge	Oui	a	b	—	—	—
Isochronous Mode Problem Notification	Indique un problème d'applications isochrones, par exemple lancées trop tard.	Obligatoire pour la prise en charge du mode isochrone	Oui	a	b	—	—	—
Network Component Problem Notification	Indique un problème de composants réseau, par exemple la trame n'a lâché aucune ressource.	Obligatoire pour la prise en charge du diagnostic de composant réseau	Oui	a	b	—	—	—
Time Data Changed Notification	Indique un changement de synchronisation du temps.	Obligatoire pour la prise en charge de la synchronisation du temps	Oui	a	b	—	—	—
Dynamic Frame Packing Notification	Indique un problème au cours de la condensation dynamique des trames	Obligatoire pour la prise en charge de la condensation dynamique de trame	Oui	a	b	—	—	—
MRPD problem Notification	Indique un problème au cours de l'opération MRPD	Facultatif pour prendre en charge la MRPD	Oui	a	b	—	—	—
System Redundancy	Indique à un appareil de commande E/S qu'il ne détient plus d'AR primaire	Obligatoire pour prendre en charge la redondance d'appareil de commande E/S	Oui	a	b	—	—	—
Multiple Interface Mismatch Notification	Indique un réglage incohérent affectant plus d'une interface	Obligatoire	Oui	a	b	—	—	—
Upload and Retrieval Notification	Indique une demande de téléchargement vers l'amont et de stockage d'objets Record Data.	Facultatif	Non	—	—	e	—	—

Alarm type	Usage	Utilisation	Attaché à l'ASE Diagnosis	Alarm Payload				
				Maintenance Item	Diagnosis Item	Upload & Retrieval iParameter Item	Manufacturer Alarm Item	Profile Alarm Item
Pull module	Doit indiquer une alarme si un module utilisé est "pulled" (tiré).	Obligatoire pour prendre en charge l'alarme de module "Pull"	Non	—	—	—	c	—
Manufacturer Specific	Les valeurs de 0x0020 à 0x007F peuvent être utilisées pour des types d'alarme spécifiques à un fabricant.	Facultatif	Non	—	—	—	c	—
Profile Specific	Doit être utilisé conformément aux "Profile Guidelines" ("Lignes directrices pour profils") si un appareil E/S se conforme à un profil spécifique d'appareil	Selon le profil pris en charge	Non	—	—	—	—	f
a L'élément de maintenance est acheminé, si l'élément de diagnostic contient au moins un diagnostic de voie (y compris un diagnostic étendu, qualifié ou spécifique à un fabricant) avec ChannelProperties.Maintenance mise à MAINTENANCE_REQUIRED ou MAINTENANCE_DEMANDED. b L'élément le diagnostic peut être omis si tout le diagnostic disparaît de la voie. c Les informations d'alarme peuvent facultativement contenir des informations spécifiques à un fabricant. d Les informations d'alarme peuvent facultativement contenir des informations spécifiques à un fabricant. cependant, cela n'est pas recommandé. e La charge utile d'alarme est étendue soit avec un Upload and Retrieval Item, soit avec un iParameter Item. f Les informations d'alarme peuvent facultativement contenir des informations spécifiques à un profil.								

7.3.5.2 Spécification de la classe Alarm

7.3.5.2.1 Modèle

Un objet Alarm est décrit par le modèle suivant:

ASE: ASE Alarm

CLASS: Alarm

CLASS ID: pas utilisé

PARENT CLASS: TOP

ATTRIBUTES:

1	(m) Attribut-clé:	Implicit
2	(m) Attribut:	Priority
3	(m) Attribut:	Alarm Priority High
3.1	(m) Attribut:	Sequence Number
4	(m) Attribut:	Alarm Priority Low
4.1	(m) Attribut:	Sequence Number

SERVICES:

1	(m) OpsService:	Alarm Notification

7.3.5.2.2 Attributs

Implicit

L'attribut-clé « Implicit » indique que l'objet Alarm est adressé de façon implicite par les services.

Priority

Cet attribut définit si l'ASE Alarm utilise une ou deux files d'attente d'alarmes. Noter que cet attribut est contrôlé par rapport à la CR alarme de la classe CRL.

Valeurs permises: la priorité d'alarme donnée par l'utilisateur est utilisée et les notifications d'alarme sont mises dans la file d'attente des alarmes hautes ou basses selon le cas, FIXED_PRIORITY - la priorité de l'utilisateur est ignorée et les notifications d'alarme sont toujours mises dans la file d'attente des alarmes basses (seule la file d'attente des alarmes basses est utilisée).

Un appareil ES configuré en tant que rôle de puits d'alarmes doit toujours prendre en charge les paramètres USER_PRIORITY et FIXED_PRIORITY. L'appareil de commande E/S configuré en tant que rôle de puits d'alarmes peut décider si cela le restreint à une ressource ou non.

Alarm Priority High

Cet attribut est constitué des attributs suivants:

Sequence Number

Cet attribut stocke le numéro de séquence de la dernière notification d'alarme envoyée au puits d'alarmes. Il est initialisé au moyen d'une valeur arbitraire et doit être incrémenté avec chaque notification d'alarme.

Dès qu'une alarme est répétée via différentes AR d'un ARSet, le même numéro de séquence doit être utilisé.

Cet attribut n'est pas utilisé dans le contexte d'une notification de redondance du système.

Type d'attribut: Unsigned16

Valeurs permises: 0x0000 – 0x7FFF

Alarm Priority Low

Cet attribut est constitué des attributs suivants:

Sequence Number

Cet attribut stocke le numéro de séquence de la dernière notification d'alarme envoyée au puits d'alarmes. Il est initialisé au moyen d'une valeur arbitraire et doit être incrémenté avec chaque notification d'alarme.

Dès qu'une alarme est répétée via différentes AR d'un ARSet, le même numéro de séquence doit être utilisé.

Cet attribut n'est pas utilisé dans le contexte d'une notification de redondance du système.

Type d'attribut: Unsigned16

Valeurs permises: 0x0000 – 0x7FFF

7.3.5.2.3 Invocation de l'objet Alarm

Chaque AR possède son propre objet Alarm.

7.3.5.3 Spécification des services Alarm**7.3.5.3.1 Alarm Notification**

Ce service est utilisé pour transférer une notification d'alarme de l'appareil E/S vers l'appareil de commande E/S et, facultativement, vice versa. Ce service doit être utilisé seulement conjointement à l'IO AR. Le Tableau 162 montre les paramètres du service.

Tableau 162 – Alarm Notification

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Alarm Priority	M	M(=)		
Alarm type	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Module Ident Number	M	M(=)		
Submodule Ident Number	M	M(=)		
Alarm Specifier	M	M(=)		
Sequence Number	M	M(=)		
Channel Diagnosis	U	U(=)		
Manufacturer Specific Diagnosis	U	U(=)		
Submodule Diagnosis State	U	U(=)		
AR Diagnosis State	U	U(=)		
Alarm Payload	U	U(=)		
Maintenance Item	U	U(=)		
User Structure Identifier	M	M(=)		
Maintenance Status	M	M(=)		
Alarm Item	M	M(=)		
User Structure Identifier	M	M(=)		
Diagnosis Item	C	C(=)		
Channel Diagnosis	C	C(=)		
List of Channel Diagnosis Data	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
Channel Error type	M	M(=)		
Ext Channel Diagnosis	C	C(=)		
List of Ext Channel Diagnosis Data	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
Channel Error type	M	M(=)		
Ext Channel Error type	M	M(=)		
Ext Channel Add Value	M	M(=)		
Qualified Channel Diagnosis	C	C(=)		
List of Qualified Channel Diagnosis Data	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
Channel Error type	M	M(=)		
Ext Channel Error type	M	M(=)		
Ext Channel Add Value	M	M(=)		
Qualified Channel Qualifier	M	M(=)		
Manufacturer Specific Diagnosis	C	C(=)		
List of Manufacturer Specific Diagnosis Data	M	M(=)		

Nom de paramètre	req	ind	rsp	cnf
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Channel Number	M	M(=)		
Channel Properties	M	M(=)		
User Structure Identifier	M	M(=)		
List of Data	M	M(=)		
Data	M	M(=)		
Upload & Retrieval Item	C	C(=)		
List of Records	M	M(=)		
Record Number	M	M(=)		
Record Length	M	M(=)		
iParameter Item	C	C(=)		
iPar Req Header	M	M(=)		
Max Segm Size	M	M(=)		
Transfer Index	M	M(=)		
Total iPar Size	M	M(=)		
Manufacturer Alarm Item	C	C(=)		
List of Data	M	M(=)		
Data	M	M(=)		
Profile Alarm Item	C	C(=)		
List of Data	M	M(=)		
Data	M	M(=)		
Result(+)			S	S
AREP			M	M
Alarm type			M	M
API			M	M
Slot Number			M	M
Subslot Number			M	M
Alarm Specifier			M	M
PNIO Status			M	M
Result(-)			S	S
AREP			M	M
PNIO Status			M	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Alarm Priority

Ce paramètre contient la priorité de l'alarme.

Valeurs permises: ALARM_HIGH, ALARM_LOW

Alarm type

Type d'attribut: Unsigned16

Valeurs permises: voir Tableau 161

API

Ce paramètre définit l'API à laquelle appartient le sous-module émettant l'alarme.

Type d'attribut: Unsigned32

Valeurs permises: 0 comme processus d'application par défaut; 1 à 0xFFFFFFFF sont réservés pour usage dans les lignes directrices des profils.

Slot Number

Ce paramètre définit le numéro de baie comme un identificateur local. Ce numéro est unique au sein d'un appareil. La numérotation des baies au sein d'un appareil doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. Cet attribut doit contenir la baie physique ou logique qui émet l'alarme.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

Subslot Number

Ce paramètre définit le numéro de sous-baie comme un identificateur local. Ce numéro est unique au sein d'un appareil. La numérotation des sous-baies au sein d'une baie doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. Cet attribut doit contenir la sous-baie physique ou logique qui émet l'alarme.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Module Ident Number

Ce paramètre contient l'identification du module.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFFFFF

NOTE 1 La valeur de paramètre est spécifique à un fabricant.

Submodule Ident Number

Ce paramètre contient l'identification du sous-module.

Type d'attribut: Unsigned32

NOTE 2 La valeur de paramètre est spécifique à un fabricant.

Alarm Specifier

Ce paramètre est constitué des paramètres suivants.

Sequence Number

Ce paramètre définit le numéro de séquence de l'alarme. Au moyen du Sequence Number, le destinataire détecte les doublons et reflète la valeur du champ dans l'Alarm.

Valeurs permises: 0 à 2 047

Channel Diagnosis

Ce paramètre ne doit être utilisé que pour des types d'Alarm liés à l'ASE Diagnosis (voir Tableau 161). Ce paramètre n'est pas valide pour tous les autres types d'alarme.

Le Tableau 163 montre les valeurs permises.

Tableau 163 – Channel Diagnosis

Valeur	Signification
NO_CHANNEL_DIAG	Signifie que le Subslot Number n'inclut pas un Channel Diagnosis.
CHANNEL_DIAG	Signifie que le Subslot Number inclut au moins un Channel Diagnosis

Manufacturer Specific Diagnosis

Ce paramètre ne doit être utilisé que pour des types d'Alarm liés à l'ASE Diagnosis (voir Tableau 161). Ce paramètre n'est pas valide pour tous les autres types d'alarme.

Le Tableau 164 montre les valeurs permises.

Tableau 164 – Manufacturer Specific Diagnosis

Valeur	Signification
NO_MAN_DIAG	Signifie que le Subslot Number n'inclut pas un Manufacturer Specific Diagnosis.
MAN_DIAG	Signifie que le Subslot Number inclut au moins un Manufacturer Specific Diagnosis.

Submodule Diagnosis State

Ce paramètre ne doit être utilisé que pour des types d'Alarm liés à l'ASE Diagnosis (voir Tableau 161). Ce paramètre n'est pas valide pour tous les autres types d'alarme.

Le Tableau 165 montre les valeurs permises.

Tableau 165 – Submodule Diagnosis State

Valeur	Signification
NO_DIAG	Sans erreur. En outre, elle indique que tous les diagnostics rapportés ont été supprimés. Une notification "disappears" individuelle peut être omise. Cependant, même dans ce cas, un diagnostic de voie traité localement comme un statut et/ou un Manufacturer Specific Diagnosis traité comme un statut peu(ven)t être présent(s).
DIAG	Il existe au moins un diagnostic au niveau du sous-module. Il pourrait y avoir un ou plusieurs Channel Diagnosis et/ou un ou plusieurs Manufacturer Specific Diagnosis.

AR Diagnosis State

Ce paramètre ne doit être utilisé que pour des types d'Alarm liés à l'ASE Diagnosis (voir Tableau 161). Ce paramètre n'est pas valide pour tous les autres types d'alarme.

Le paramètre AR Diagnosis State doit correspondre à l'état courant du paramètre Problem Indicator Flag de l'ASE «IO Data ASE», qui est relatif à tous les modules/sous-modules qui appartiennent à l'AR.

Le Tableau 166 montre les valeurs permises.

Tableau 166 – AR Diagnosis State

Valeur	Signification
NO_DIAG	Sans erreur. De plus, elle indique que tous les diagnostics rapportés qui sont relatifs à l'AR ont été supprimés. Une notification "disappears" individuelle peut être omise. Cependant, même dans ce cas, un diagnostic de voie (Channel Diagnosis) traité localement comme un statut et/ou un Manufacturer Specific Diagnosis traité comme un statut peu(ven)t être présent(s).
DIAG	Au moins un diagnostic existe en un sous-module lié à l'AR. Il pourrait y avoir un ou plusieurs Channel Diagnosis et/ou un ou plusieurs Manufacturer Specific Diagnosis.

Alarm Payload

Ce paramètre facultatif contient la charge utile d'alarme; structurée selon les règles indiquées dans le Tableau 161. Il est constitué des paramètres suivants:

Maintenance Item

Ce paramètre facultatif contient le statut de la maintenance. Il ne doit être présent que conjointement aux notifications d'alarme liées à l'ASE de diagnostic et si au moins un des paramètres en dessous possède des informations qualifiées ou de maintenance. Autrement, il doit être omis

Il est constitué des paramètres suivants:

User Structure Identifier

Ce paramètre identifie le format de l'élément.

Type d'attribut: Unsigned16

Valeurs permises: MAINTENANCE_USI

Maintenance Status

Ce paramètre contient le statut de la maintenance du sous-module. Il est constitué des paramètres suivants:

Maintenance Required

Ce paramètre contient la valeur MAINTENANCE_REQUIRED si au moins une voie de la source d'alarme (API, Slot, Subslot) contient des informations de maintenance requise.

Valeurs permises: MAINTENANCE_REQUIRED, NO_MAINTENANCE_REQUIRED

Maintenance Demanded

Ce paramètre contient la valeur MAINTENANCE_DEMANDED si au moins une voie de la source d'alarme (API, Slot, Subslot) contient des informations de maintenance exigée.

Valeurs permises: MAINTENANCE_DEMANDED, NO_MAINTENANCE_DEMANDED

Qualifier 2

Ce paramètre contient la valeur QUALIFIER_2_SET si au moins une voie de la source d'alarme (API, Slot, Subslot) contient des informations qualifiées. La sémantique doit être définie par des profils.

Valeurs permises: QUALIFIER_2_SET, QUALIFIER_2_NOT_SET

...

Qualifier 31

Ce paramètre contient la valeur QUALIFIER_31_SET si au moins une voie de la source d'alarme (API, Slot, Subslot) contient des informations qualifiées. La sémantique doit être définie par des profils.

Valeurs permises: QUALIFIER_31_SET, QUALIFIER_31_NOT_SET

Alarm Item

Ce paramètre facultatif contient les informations correspondantes pour l'alarme. Il est présent, si de plus amples informations sont transmises. Les règles selon lesquelles l'élément (à savoir, l'élément de diagnostic, de téléchargement et de récupération, iParamètre ou d'alarme commune) est transmis, sont décrites dans le Tableau 161.

Ce paramètre est constitué des paramètres suivants:

User Structure Identifier

Ce paramètre identifie le format de l'élément.

Type d'attribut: Unsigned16

Valeurs permises: indiquées dans le Tableau 167.

Tableau 167 – User Structure Identifier

Valeur	Signification / Usage
CHANNEL_DIAGNOSIS_USI	Doit être utilisé seulement conjointement à Channel Diagnosis dans Alarm Notification et Diagnosis Data.
EXT_CHANNEL_DIAGNOSIS_USI	Doit être utilisé seulement conjointement à Extended Channel Diagnosis dans Alarm Notification et Diagnosis Data.
QUALIFIED_CHANNEL_DIAGNOSIS_USI	Doit être utilisé seulement conjointement à Qualified Channel Diagnosis dans Alarm Notification et Diagnosis Data.
MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF	Conjointement au type d'alarme Diagnosis, Manufacturer Specific Diagnosis dans Alarm Notification et Diagnosis Data. Conjointement avec d'autres types d'alarme, l'usage est spécifique à un fabricant.
PROFILE_SPECIFIC_USI_0x9000 à PROFILE_SPECIFIC_USI_0x9FFF	La signification est définie dans les spécifications de profils.
UPLOAD_AND_RETRIEVAL_USI	Doit être utilisé conjointement à une notification "Upload and Retrieval".
IPARAMETER_USI	Doit être utilisé conjointement à une notification iParameter.

Diagnosis Item

Ce paramètre facultatif n'est présent que si l'User Structure Identifier est mis sur CHANNEL_DIAGNOSIS_USI, EXT_CHANNEL_DIAGNOSIS_USI, QUALIFIED_CHANNEL_DIAGNOSIS_USI, MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF.

Un élément de liste est constitué des paramètres suivants:

Channel Diagnosis

Ce paramètre facultatif n'est présent que si User Structure Identifier est mis sur CHANNEL_DIAGNOSIS_USI. Il est constitué des paramètres suivants:

List of Channel Diagnosis Data

Ce paramètre contient une liste de donnée de diagnostic de voie. Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre définit le numéro de voie comme un identificateur local. Ce numéro est unique au sein d'une sous-baie. La numérotation des voies au sein d'une sous-baie doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. La liste doit contenir toutes les voies physiques ou logiques qui ont un diagnostic.

La valeur spéciale 0x8000 identifie l'ensemble du sous-module, au lieu d'une voie particulière.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF numéro de voie spécifique à un utilisateur, 0x8000 sous-module

Channel Properties

Ce paramètre est constitué des paramètres suivants:

Type

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Accumulative

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Maintenance

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Spécifier (spécificateur)

Ce paramètre a une définition étendue par rapport à celle spécifiée dans l'ASE Diagnosis.

Ce paramètre indique que le diagnostic/la maintenance apparaît ou disparaît. La sémantique est indiquée dans le Tableau 168.

Valeurs permises: APPEARS, DISAPPEARS, DISAPPEARS_BUT_OTHER_REMAIN, ALL_DISAPPEARS

Tableau 168 – Sémantique de Spécifier

Maintenance	Spécifier (spécificateur)	Signification
DIAGNOSIS	APPEARS	Diagnosis apparaît. L'ASE Diagnosis contient cette entrée et éventuellement d'autres entrées avec ChannelProperties.Maintenance = DIAGNOSIS pour cette voie.
DIAGNOSIS	DISAPPEARS	Diagnosis disparaît L'ASE Diagnosis ne contient aucune autre entrée avec ChannelProperties.Maintenance = DIAGNOSIS pour cette voie.
DIAGNOSIS	DISAPPEARS_BUT_OTHER_REMAIN	Diagnosis disparaît L'ASE Diagnosis contient encore d'autres entrées avec ChannelProperties.Maintenance = DIAGNOSIS pour cette voie.
DIAGNOSIS	ALL_DISAPPEARS	Diagnosis disparaît L'ASE Diagnosis ne contient plus d'entrées pour cette voie, toutes les informations ^a ultérieures de Diagnosis, Maintenance (requis et exigé) et Qualified disparaissent.
MAINTENANCE_REQUIRED	APPEARS	Maintenance Required appears L'ASE Diagnosis contient cette entrée et éventuellement d'autres entrées avec ChannelProperties.Maintenance = MAINTENANCE_REQUIRED pour cette voie.
MAINTENANCE_REQUIRED	DISAPPEARS	Maintenance Required disparaît L'ASE Diagnosis ne contient aucune autre entrée avec ChannelProperties.Maintenance = MAINTENANCE_REQUIRED pour cette voie.

Maintenance	Spécifier (spécificateur)	Signification
MAINTENANCE_REQUIRED	DISAPPEARS_BUT_OTHER_REMAIN	Maintenance Required disparaît L'ASE Diagnosis contient encore d'autres entrées avec ChannelProperties.Maintenance = MAINTENANCE_REQUIRED pour cette voie.
MAINTENANCE_DEMANDED	APPEARS	Maintenance Demanded disparaît L'ASE Diagnosis contient cette entrée et éventuellement d'autres entrées avec ChannelProperties.Maintenance = MAINTENANCE_DEMANDED pour cette voie.
MAINTENANCE_DEMANDED	DISAPPEARS	Maintenance Demanded disparaît L'ASE Diagnosis ne contient aucune autre entrée avec ChannelProperties.Maintenance = MAINTENANCE_DEMANDED pour cette voie.
MAINTENANCE_DEMANDED	DISAPPEARS_BUT_OTHER_REMAIN	Maintenance Demanded disparaît L'ASE Diagnosis contient encore d'autres entrées avec ChannelProperties.Maintenance = MAINTENANCE_DEMANDED pour cette voie.
QUALIFIED_DIAGNOSIS	APPEARS	Qualified Diagnosis disparaît L'ASE Diagnosis contient cette entrée et éventuellement d'autres entrées avec ChannelProperties.Maintenance = QUALIFIED_DIAGNOSIS pour cette voie.
QUALIFIED_DIAGNOSIS	DISAPPEARS	Qualified Diagnosis disparaît L'ASE Diagnosis ne contient aucune autre entrée avec ChannelProperties.Maintenance = QUALIFIED_DIAGNOSIS pour cette voie.
QUALIFIED_DIAGNOSIS	DISAPPEARS_BUT_OTHER_REMAIN	Qualified Diagnosis disparaît L'ASE Diagnosis contient encore d'autres entrées avec ChannelProperties.Maintenance = QUALIFIED_DIAGNOSIS pour cette voie.
^a "Ultérieurs" signifie que, pour ce ChannelErrorType particulier, tous les ExtChannelErrorTypes (et le ChannelErrorType lui-même) avec diagnostic, maintenance requise, maintenance exigée, et données qualifiées changent en OK.		

Sens

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Channel Error type

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Ext Channel Diagnosis

Ce paramètre facultatif est présent si l'attribut User Structure Identifier est mis sur EXT_CHANNEL_DIAGNOSIS_USI. Il est constitué des paramètres suivants:

List of Ext Channel Diagnostic Data

Ce paramètre contient une liste de donnée de diagnostic de voie étendu. Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Channel Properties

Ce paramètre a la même définition que celle spécifiée dans le Channel Diagnosis ci-dessus.

Channel Error type

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Ext Channel Error type

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Ext Channel Add Value

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Qualified Channel Diagnosis

Ce paramètre facultatif est présent si l'attribut User Structure Identifier est mis sur QUALIFIED_CHANNEL_DIAGNOSIS_USI. Il est constitué des paramètres suivants:

List of Qualified Channel Diagnostic Data

Ce paramètre contient une liste de donnée de diagnostic de voie étendu. Un élément de liste est constitué des paramètres suivants:

Channel Number

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Channel Properties

Ce paramètre a la même définition que celle spécifiée dans le Channel Diagnosis ci-dessus.

Channel Error type

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Ext Channel Error type

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Ext Channel Add Value

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Qualified Channel Qualifier

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Manufacturer Specific Diagnosis

Ce paramètre facultatif est présent si le User Structure Identifier est mis sur MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF. Il est constitué des paramètres suivants:

Les structures spécifiques à un fabricant doivent être utilisées seulement si Channel Diagnosis ou Extended Channel Diagnosis ou Qualified Extended Channel Diagnosis ne couvre pas les informations de diagnostic.

List of Manufacturer Specific Diagnostic Data

Ce paramètre contient une liste de données de diagnostic de voie spécifiques à un fabricant. Un élément de liste est constitué des paramètres suivants:

API

Ce paramètre définit l'API à laquelle appartient le sous-module émettant l'alarme.

Type d'attribut: Unsigned32

Valeurs permises: 0 comme processus d'application par défaut; 1 à 0xFFFFFFFF sont réservés pour usage dans les lignes directrices des profils.

Slot Number

Ce paramètre définit le numéro de baie comme un identificateur local. Ce numéro est unique au sein d'un appareil. La numérotation des baies au sein d'un appareil doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. Cet attribut doit contenir la baie physique ou logique qui émet l'alarme.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

Subslot Number

Ce paramètre définit le numéro de sous-baie comme un identificateur local. Ce numéro est unique au sein d'un appareil. La numérotation des sous-baies au sein d'une baie doit être dans l'ordre croissant, mais elle peut contenir des trous afin de permettre une structuration logique. Cet attribut doit contenir la sous-baie physique ou logique qui émet l'alarme.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Channel Number

Ce paramètre a la même définition que celle spécifiée dans l'ASE Diagnosis.

Channel Properties

Ce paramètre a la même définition que celle spécifiée dans le Channel Diagnosis ci-dessus.

User Structure Identifier

Ce paramètre indique la structure du paramètre Manufacturer Specific Diagnosis Data.

Type d'attribut: Unsigned16

Valeurs permises: MANUFACTURER_SPECIFIC_USI_0x0000 à MANUFACTURER_SPECIFIC_USI_0x7FFF

List of Data

Ce paramètre contient une liste de données de diagnostic spécifique à un fabricant. Un élément de liste est constitué des paramètres suivants:

Data

Ce paramètre contient les données de diagnostic spécifique à un fabricant.

Type d'attribut: L'un des types de données suivants doit être utilisé: Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference, Floating Point, Network Time, VisibleString, Network Time Difference, OctetString

Upload and Retrieval Item

Ce paramètre facultatif n'est présent que si Alarm Type est mis à Upload and Retrieval Notification» et si le User Structure Identifier est mis à UPLOAD_AND RETRIEVAL_USI.

Avec ce paramètre, un sous-module indique l'existence d'objets Record Data qui doivent être stockés à l'appareil de commande E/S. L'appareil de commande E/S doit télécharger vers l'amont et stocker en permanence les objets Record Data. A chaque séquence de connexion suivante et pour chaque séquence suivante de branchement et libération, l'appareil de commande E/S doit acheminer ces objets Record Data supplémentaires vers la source des objets Record Data stockés. Les objets Record Data qui font partie de la description GSD et sont acheminés du système d'ingénierie vers l'appareil de commande E/S ne doivent pas faire partie de la List of Records.

Il est constitué des paramètres suivants:

List of Records

Ce paramètre contient une liste d'objets Record Data. Un élément de liste est constitué des paramètres suivants:

Record Number

Ce paramètre contient l'Index d'un objet Record Data tel que montré dans l'ASE Record Data.

Valeurs permises: 0 à 0xFFFF

Record Length

Ce paramètre contient la Length d'un objet Record Data tel que montré dans l'ASE Record Data.

Valeurs permises: 2^0 à $2^{32}-256$

iParameter Item

Ce paramètre facultatif n'est présent que si Alarm Type est mis sur «Upload and Retrieval Notification» et si le User Structure Identifier est mis sur IPARAMETER_USI.

Avec ce paramètre, un sous-module indique de télécharger vers l'amont les iParameters ou demande de télécharger vers l'aval les iParameters précédemment stockés.

La différence entre cet iParameter Item et le Upload and Retrieval Item défini ci-dessus est que le iParameter Item utilise un élément des enregistrements de données, permet le transfert segmenté de l'élément des enregistrements de données et transfère les données d'enregistrement sur demande seulement.

Si le sous-module demande un téléchargement vers l'amont, l'appareil de commande E/S doit télécharger et stocker de façon permanente l'objet «Record Data» indiqué dans l'alarme.

Si le sous-module demande un téléchargement vers l'aval, l'objet «Record Data» précédemment stocké (résultat d'un précédent téléchargement) est transféré au sous-module.

Les objets Record Data qui font partie de la description GSD et sont acheminés du système d'ingénierie vers l'appareil de commande E/S ne doivent pas faire partie de l'objet «Record Data» de ce iParameter.

Il est constitué des paramètres suivants:

iPar Req Header

Ce paramètre contient l'en-tête de demande utilisé pour le téléchargement vers l'amont/l'aval des iParameters. Il est constitué des paramètres suivants:

SR_Version

Ce paramètre indique la version du protocole iParameter.

Type d'attribut: Unsigned8

Valeurs permises: 0x01

N_Count

Ce paramètre indique le nombre de notifications du protocole iParameter.

Type d'attribut: Unsigned8

Valeurs permises: 0x01

SR_Type

Ce paramètre indique le sous-type de notification d'iParameter.

Type d'attribut: Unsigned8

Valeurs permises: 0x01 pour demander un téléchargement vers l'amont, 0x03 pour demander un téléchargement vers l'aval

Max Segm Size

Ce paramètre contient la taille maximale du segment à utiliser pour lire une lecture individuelle des données d'enregistrement. Il permet ainsi la fragmentation des grands iParameters.

Type d'attribut: Unsigned32

Valeurs permises: 2^0 à $2^{22}-1$

Transfer Index

Ce paramètre contient l'indice des données d'enregistrement utilisé pour le téléchargement vers l'amont/l'aval des iParameters.

Type d'attribut: Unsigned32

Valeurs permises: 0 – 0x7FFF

Total iPar Size

Ce paramètre contient la taille totale des iParameters. Ce paramètre doit être mis à la taille des iParameters lors d'une demande de téléchargement vers l'amont. Sur demande de téléchargement aval, la valeur "0" est autorisée; ce qui indique de télécharger l'ensemble du jeu de paramètres stocké dans l'appareil de commande E/S.

Type d'attribut: Unsigned32

Valeurs permises: 2^0 à $2^{32}-256$

Manufacturer Alarm Item

Ce paramètre facultatif n'est présent que si l'Alarm Type est défini sur «Manufacturer Specific» (spécifique à un fabricant). Dans ce cas le User Structure Identifier doit être mis à MANUFACTURER_SPECIFIC_USI_0x0000 jusqu'à MANUFACTURER_SPECIFIC_USI_0x7FFF.

List of Data

Ce paramètre contient une liste de données d'alarme spécifique à un fabricant.

La longueur des données utilisateur ne doit pas dépasser 1 408 octets.

Un élément de liste est constitué des paramètres suivants:

Data

Ce paramètre contient les données d'alarme spécifique à un fabricant.

Type d'attribut: L'un des types de données suivants doit être utilisé:

Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference, Floating Point, Network Time, VisibleString, Network Time Difference, OctetString

Profile Alarm Item

Ce paramètre facultatif n'est présent que si l'Alarm Type est défini sur «Profile Specific» (spécifique à un profil). Dans ce cas, User Structure Identifier doit être mis à une valeur de PROFILE_SPECIFIC_USI_0x9000 à PROFILE_SPECIFIC_USI_0x9FFF.

List of Data

Ce paramètre contient une liste de données d'alarme spécifique à un profil.

La longueur des données utilisateur ne doit pas dépasser 1 408 octets.

Un élément de liste est constitué des paramètres suivants:

Data

Ce paramètre contient les données d'alarme spécifique à un fabricant.

Type d'attribut: L'un des types de données suivants doit être utilisé:

Boolean, BinaryDate, Integer, Time of Day, Unsigned, Time-Difference, Floating Point, Network Time, VisibleString, Network Time Difference, OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi.

Alarm type

Ce paramètre reflète la valeur de la demande.

API

Ce paramètre reflète la valeur de la demande.

Slot Number

Ce paramètre reflète la valeur de la demande.

Subslot Number

Ce paramètre reflète la valeur de la demande.

Alarm Specifier

Ce paramètre reflète la valeur de la demande.

PNIO Status

Ce paramètre contient le statut d'erreur.

Valeurs permises: NO_ERROR, ALARM_TYPE_NOT_SUPPORTED, WRONG_SUBMODULE_STATE, BACKUP_ALARM_NOT_EXECUTED

Type d'attribut: Unsigned32

Result(-)

Ce paramètre indique que la demande de service a échoué.

PNIO Status

Ce paramètre contient la cause de la défaillance.

7.3.5.4 Comportement des objets Alarm

7.3.5.4.1 Comportement général du puits d'alarmes

Le comportement d'application est basé sur le contrôle de flux de l'émission d'alarmes, l'IOPS, l'IOCS et le Problem Indicator de l'ASE IO Data. Ces mécanismes rendent commode le traitement de l'alarme par l'application.

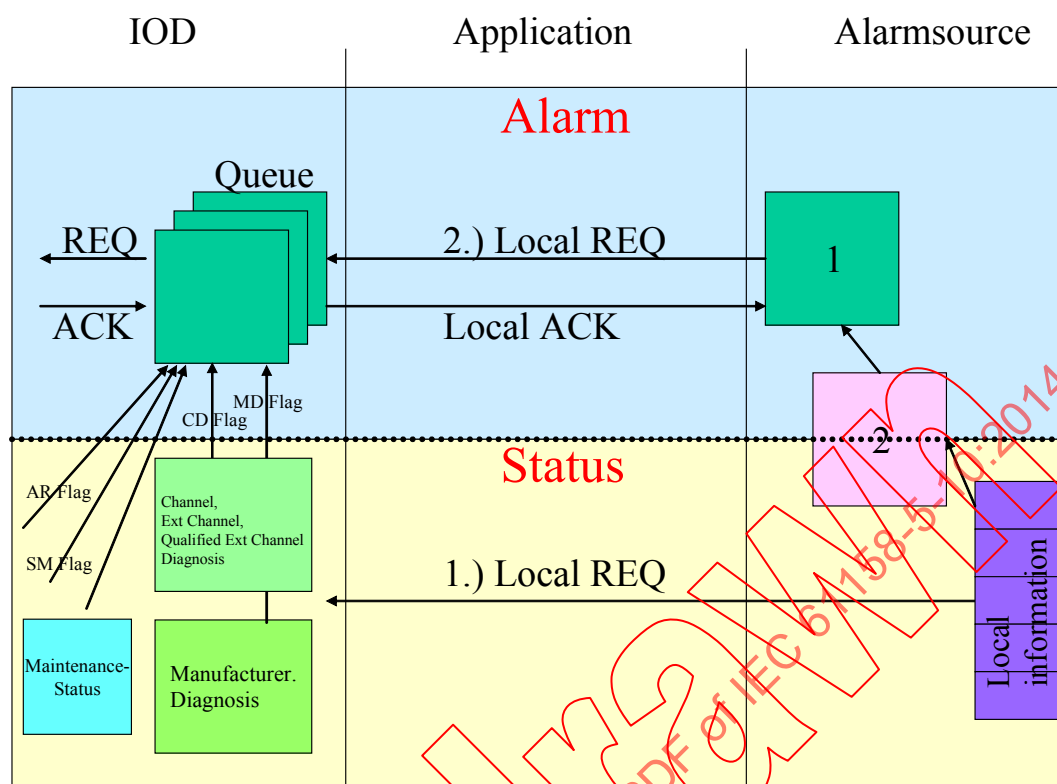
NOTE Le processus d'application peut ne pas traiter l'alarme immédiatement en interrompant d'autres processus. Il peut également rechercher par sondage des alarmes placées en file d'attente pour les traiter dans une fenêtre temporelle appropriée.

Chaque processus d'application doit maintenir une file d'attente de réception pour chaque type d'alarme. Si elle est utilisée, la taille de la file d'attente est déterminée par les mécanismes de contrôle de flux. Avec le traitement de l'alarme, l'acquittement doit être envoyé à la source pour permettre l'envoi d'autres alarmes de ce type.

Les files d'attente d'alarmes doivent être vidées en cas d'arrêt d'AR et d'alarmes "pull/plug".

7.3.5.4.2 Comportement général de la source d'alarmes

La Figure 32 montre le modèle d'une source d'alarmes. La source est divisée en l'appareil E/S (IOD), le processus d'application et la source d'alarmes.



Légende

Anglais	Français
Application	Application
IOD	IOD (appareil E/S)
Alarmsource	Source d'alarme
Queue	Queue (File d'attente)
REQ	DEMANDE
ACK	ACQUITTEMENT
Alarm	Alarme
Status	Statut
Manufacturer. Diagnosis	Diagnostic fabricant
Local information	Informations locales
2.) Local REQ	2.) DEMANDE locale
Local ACK	ACQUITTEMENT local
Channel, Ext Channel, Qualified Ext Channel, Diagnosis	Channel, Ext Channel, Qualified Ext Channel, Diagnosis
CD Flag	Fanion CD
MD Flag	Fanion MD
1.) Local REQ	1.) DEMANDE locale
AR Flag	Fanion AR
SM Flag	Fanion SM
Maintenance-Status	Statut de maintenance

Figure 32 – Exemple de modèle de ressources en une source d'alarmes

Le modèle montre comment traiter un surdébit d'événements à de la source. Le surdébit d'événements peut être provoqué par le contrôle de flux, car le puits détermine la rapidité avec laquelle les alarmes sont traitées. Un mélange de modèles de file d'attente et de tampon résout ce problème pour l'ASE Alarm.

La source d'alarme maintient des informations locales relatives à l'état. Si, par exemple, le statut change de GOOD en BAD, cela est notifié. Elle maintient au moins deux tampons pour les alarmes. Le premier tampon est utilisé pour transporter la transition d'état. Le second tampon contient l'état courant. Une nouvelle transition d'état ne sera pas rapportée tant que l'acquiescement n'est pas reçu dans le cas de deux tampons seulement. Mais l'état courant peut être lu à tout moment avec des informations de diagnostic.

L'appareil E/S gère deux files d'attente, une pour la priorité ALARM_HIGH et une pour la priorité ALARM_LOW. L'appareil de commande E/S peut demander via la CR alarme que seule la file d'attente des alarmes de priorité ALARM_LOW soit utilisée par l'appareil E/S.

Une nouvelle alarme de la même priorité est bloquée tant que l'acquiescement est en cours. Le fait de bloquer les alarmes de même priorité en attendant l'Alarm_ACK de l'appareil de commande E/S entraîne la mise en attente des alarmes additionnelles dans la file d'attente des messages au niveau de l'appareil E/S. Cette file d'attente peut comporter des entrées qui sont désuètes du fait d'autres entrées (ultérieures) (ex: diagnostic de « court-circuit » à venir ou en cours en cas de câble desserré). Dans un tel cas, l'appareil E/S peut supprimer les entrées à des fins d'optimisation à condition que cette suppression n'affecte pas l'utilisateur.

En outre, les files d'attente d'alarmes doivent être vidées en cas d'arrêt d'AR et d'alarmes "pull/plug".

7.3.5.4.3 Comportement pour des changements de diagnostic

Les alarmes de diagnostic sont le moyen de rapporter et d'acquiescer des changements de diagnostic. Il est fortement recommandé d'utiliser Channel Diagnoses ou Extended Channel Diagnosis en raison de la structure bien définie.

La séquence recommandée pour signaler un diagnostic est:

- En premier lieu, mettre à jour l'ASE "Diagnosis"
- En second lieu, émettre une alarme.

Les alarmes de diagnostic incluent des informations de la source d'alarmes et du sens des changements. L'attribut Alarm Specifier contient des informations relatives au sens. Par exemple: "appears" (apparaît) ou "disappears" (disparaît).

"Tous les diagnostics de voie ont disparu" est rapporté par une alarme avec l'attribut Submodule Diagnosis State "NO_ERROR" du paramètre Alarm Specifier. Il peut sauvegarder un grand nombre de messages individuels.

7.3.6 ASE Context

7.3.6.1 Vue d'ensemble

L'ASE Context fournit un jeu de services pour établir des relations d'application, paramétrer et configurer l'AP d'un appareil E/S lui-même et ses baies/sous-baies contenant des données E/S.

Le processus d'application des appareils E/S peut avoir besoin d'un paramètre (la plage d'un signal analogique, par exemple) provenant de l'appareil de commande E/S assigné pour fournir les données issues de l'objet Input Data et/ou de l'objet Output Data conformément aux exigences de l'AP de l'appareil de commande E/S. Ces appareils E/S fournissent dans l'ASE Context un objet IO AR Parameter ou, pour plus de flexibilité, plusieurs objets IO AR

Structured Parameter. Les valeurs pour les attributs de ces objets seront fournies par l'appareil de commande E/S et assignées après validation de la part de l'AP de l'appareil E/S.

La fonctionnalité d'une application d'appareil E/S implique les nécessaires objets de paramétrisation (objets Record Data spécifique) qui doivent être acheminés, à la phase de démarrage du système, de l'appareil de commande E/S vers les appareils E/S assignés par le biais de l'IO AR. L'application de l'appareil E/S doit vérifier les données de paramétrisation envoyées par l'appareil de commande E/S.

NOTE En comparaison à l'ASE IO Data, dans lequel toutes les baies et sous-baies réelles avec leurs attributs sont stockées, l'ASE Context contient seulement les baies ou sous-baies configurées ou utilisées, après un établissement réussi de la connexion.

7.3.6.2 Spécification de la classe Context

7.3.6.2.1 Modèle

Cette classe spécifie l'objet pour les données de paramètre qui sont relatives à l'appareil E/S lui-même et/ou aux baies/sous-baies de l'appareil E/S.

L'objet IO Device context est décrit par le modèle suivant:

ASE:	ASE Context
CLASS:	Contexte
CLASS ID:	pas utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1	(m) Attribut-clé: Implicit
2	(m) Attribut: List of APs
2.1	(m) Attribut: API
2.2	(m) Attribut: Real Identification
2.2.1	(m) Attribut: List of Slots
2.2.1.1	(m) Attribut: Slot Number
2.2.1.2	(m) Attribut: Module Ident Number
2.2.1.3	(m) Attribut: Module Properties
2.2.1.4	(m) Attribut: List of Subslots
2.2.1.4.1	(m) Attribut: Subslot Number
2.2.1.4.2	(m) Attribut: Submodule Ident Number
2.2.1.4.3	(m) Attribut: Submodule Properties
2.2.1.4.3.1	(m) Attribut: Type
2.2.1.4.3.2	(m) Attribut: Shared Input
2.2.1.4.3.3	(m) Attribut: Reduce Input Submodule Data Length
2.2.1.4.3.4	(m) Attribut: Reduce Output Submodule Data Length
2.2.1.4.3.5	(m) Attribut: Discard IOXS
2.2.1.4.4	(m) Attribut: List of Data Descriptions
2.2.1.4.4.1	(m) Attribut: Data Direction
2.2.1.4.4.2	(m) Attribut: Submodule Data Length
2.2.1.4.4.3	(m) Attribut: Length IOPS
2.2.1.4.4.4	(m) Attribut: Length IOCS
2.3	(m) Attribut: Expected Identification
2.3.1	(m) Attribut: List of ARs
2.3.1.1	(m) Attribut: AREP
2.3.1.2	(o) Attribut: Sync State
2.3.1.3	(m) Attribut: List of CRs
2.3.1.3.1	(m) Attribut: CREP
2.3.1.3.2	(m) Attribut: Multicast Provider State

2.3.1.4	(m) Attribut:	List of APIs
2.3.1.4.1	(m) Attribut:	API
2.3.1.4.2	(m) Attribut:	List of Slots
2.3.1.4.2.1	(m) Attribut:	Slot Number
2.3.1.4.2.2	(m) Attribut:	Module Ident Number
2.3.1.4.2.3	(m) Attribut:	Module Properties
2.3.1.4.2.4	(m) Attribut:	Module State
2.3.1.4.2.5	(m) Attribut:	List of Subslots
2.3.1.4.2.5.1	(m) Attribut:	Subslot Number
2.3.1.4.2.5.2	(m) Attribut:	Submodule Ident Number
2.3.1.4.2.5.3	(m) Attribut:	Submodule Properties
2.3.1.4.2.5.3.1	(m) Attribut:	Type
2.3.1.4.2.5.3.2	(m) Attribut:	Shared Input
2.3.1.4.2.5.4	(m) Attribut:	Submodule State
2.3.1.4.2.5.4.1	(m) Attribut:	Format Indicator
2.3.1.4.2.5.4.2	(c) Contrainte:	Format Indicator = V_1_0
2.3.1.4.2.5.4.2.1	(m) Attribut:	Detail
2.3.1.4.2.5.4.3	(c) Contrainte:	Format Indicator <> V_1_0
2.3.1.4.2.5.4.3.1	(m) Attribut:	Add Info
2.3.1.4.2.5.4.3.2	(m) Attribut:	Qualifier Info
2.3.1.4.2.5.4.3.3	(m) Attribut:	Maintenance Required
2.3.1.4.2.5.4.3.4	(m) Attribut:	Maintenance Demanded
2.3.1.4.2.5.4.3.5	(m) Attribut:	Diag Info
2.3.1.4.2.5.4.3.6	(m) Attribut:	AR Info
2.3.1.4.2.5.4.3.7	(m) Attribut:	Ident Info
3	(m) Attribut:	I&M Data
3.1	(m) Attribut:	List of Slots
3.1.1	(m) Attribut:	Slot Number
3.1.2	(m) Attribut:	List of Subslots
3.1.2.1	(m) Attribut:	Subslot Number
3.1.2.2	(m) Attribut:	I&M0 Data
3.1.2.2.1	(m) Attribut:	Vendor ID (Identificateur du vendeur)
3.1.2.2.2	(m) Attribut:	Order ID
3.1.2.2.3	(m) Attribut:	Serial Number (Numéro de série)
3.1.2.2.4	(m) Attribut:	Hardware Revision
3.1.2.2.5	(m) Attribut:	Software Revision
3.1.2.2.5.1	(m) Attribut:	Prefix
3.1.2.2.5.2	(m) Attribut:	Functional Enhancement
3.1.2.2.5.3	(m) Attribut:	Bug Fix
3.1.2.2.5.4	(m) Attribut:	Internal Change
3.1.2.2.6	(m) Attribut:	Revision Counter
3.1.2.2.7	(m) Attribut:	Profile ID
3.1.2.2.8	(m) Attribut:	Profile Specific Type
3.1.2.2.9	(m) Attribut:	I&M Version
3.1.2.2.10	(m) Attribut:	I&M Supported
3.1.2.3	(o) Attribut:	I&M1 Data
3.1.2.3.1	(m) Attribut:	Fonction
3.1.2.3.2	(m) Attribut:	Location
3.1.2.4	(o) Attribut:	I&M2 Data
3.1.2.4.1	(m) Attribut:	Installation Date
3.1.2.5	(o) Attribut:	I&M3 Data
3.1.2.5.1	(m) Attribut:	Description
3.1.2.6	(o) Attribut:	I&M4 Data

3.1.2.6.1	(m) Attribut:	Signature
4	(m) Attribut:	I&M0 Filter Data
4.1	(m) Attribut:	I&M0 Carrier Data
4.1.1	(m) Attribut:	List of APs
4.1.1.1	(m) Attribut:	API
4.1.1.2	(m) Attribut:	List of Slots
4.1.1.2.1	(m) Attribut:	Slot Number
4.1.1.2.2	(m) Attribut:	Module Ident Number
4.1.1.2.3	(m) Attribut:	List of Subslots
4.1.1.2.3.1	(m) Attribut:	Subslot Number
4.1.1.2.3.2	(m) Attribut:	Submodule Ident Number
4.2	(o) Attribut:	Module Representative Data
4.2.1	(m) Attribut:	List of APs
4.2.1.1	(m) Attribut:	API
4.2.1.2	(m) Attribut:	List of Slots
4.2.1.2.1	(m) Attribut:	Slot Number
4.2.1.2.2	(m) Attribut:	Module Ident Number
4.2.1.2.3	(m) Attribut:	List of Subslots
4.2.1.2.3.1	(m) Attribut:	Subslot Number
4.2.1.2.3.2	(m) Attribut:	Submodule Ident Number
4.3	(m) Attribut:	Device Representative Data
4.3.1	(m) Attribut:	List of APs
4.3.1.1	(m) Attribut:	API
4.3.1.2	(m) Attribut:	List of Slots
4.3.1.2.1	(m) Attribut:	Slot Number
4.3.1.2.2	(m) Attribut:	Module Ident Number
4.3.1.2.3	(m) Attribut:	List of Subslots
4.3.1.2.3.1	(m) Attribut:	Subslot Number
4.3.1.2.3.2	(m) Attribut:	Submodule Ident Number
5	(o) Attribut:	Autoconfiguration Data
5.1	(m) Attribut:	Communication Data
5.1.1	(m) Attribut:	Min Device Interval
5.1.2	(m) Attribut:	Communication Properties
5.1.2.1	(m) Attribut:	DFP
5.1.2.2	(m) Attribut:	RTC3
5.1.3	(m) Attribut:	Send Clock Factors
5.2	(m) Attribut:	Configuration Data
5.2.1	(m) Attribut:	List of Offered Submodules
5.2.1.1	(m) Attribut:	API
5.2.1.2	(m) Attribut:	Slot Number
5.2.1.3	(m) Attribut:	Subslot Number
5.2.1.4	(m) Attribut:	Module Ident Number
5.2.1.5	(m) Attribut:	Submodule Ident Number
5.2.1.6	(m) Attribut:	Length Input Data
5.2.1.7	(m) Attribut:	Length Output Data
5.3	(o) Attribut:	Isochronous Data
5.3.1	(m) Attribut:	List of Offered Submodules
5.3.1.1	(m) Attribut:	API
5.3.1.2	(m) Attribut:	Slot Number
5.3.1.3	(m) Attribut:	Subslot Number
5.3.1.4	(m) Attribut:	Time IO Base
5.3.1.5	(m) Attribut:	Time IO Input
5.3.1.6	(m) Attribut:	Time IO Output

6	(o) Attribut:	Fast Startup Data
6.1	(m) Attribut:	List of ARs
6.1.1	(m) Attribut:	AREP
6.1.2	(m) Attribut:	Expected Fast Startup Data
6.1.2.1	(o) Attribut:	FS Parameter Block
6.1.2.1.1	(m) Attribut:	FS Parameter Mode
6.1.2.1.2	(m) Attribut:	FS Parameter UUID
6.1.2.2	(o) Attribut:	Manufacturer Specific Fast Startup Data
7	(o) Attribut:	GSD Data

SERVICES:

1	(m) OpsService:	Connect
2	(m) OpsService:	Connect Device Access
3	(m) OpsService:	Release
4	(o) OpsService:	Prm Begin
5	(m) OpsService:	Prm End
6	(m) OpsService:	Application Ready
7	(m) OpsService:	Read Expected Identification Data for one subslot
8	(m) OpsService:	Read Real Identification Data for one subslot
9	(m) OpsService:	Read Expected Identification Data for one slot
10	(m) OpsService:	Read Real Identification Data for one slot
11	(m) OpsService:	Read Expected Identification Data for one AR
12	(m) OpsService:	Read Real Identification Data for one AR
13	(m) OpsService:	Read Real Identification Data for one API
14	(m) OpsService:	Read Module Diff Block
15	(m) OpsService:	Abort
16	(m) OpsService:	Local AR Abort
17	(m) OpsService:	Read AP Data
18	(m) OpsService:	Read I&M0 Filter Data
19	(m) OpsService:	Read I&M0 Data
20	(o) OpsService:	Write I&M1 Data
21	(o) OpsService:	Read I&M1 Data
22	(o) OpsService:	Write I&M2 Data
23	(o) OpsService:	Read I&M2 Data
24	(o) OpsService:	Write I&M3 Data
25	(o) OpsService:	Read I&M3 Data
26	(o) OpsService:	Write I&M4 Data
27	(o) OpsService:	Read I&M4 Data
28	(o) OpsService:	Read Autoconfiguration Data
29	(o) OpsService:	Write Expected Fast Startup Data
30	(o) OpsService:	Read Expected Fast Startup Data
31	(o) OpsService:	Read GSD Data

7.3.6.2.2 Attributs

Implicit

L'attribut clé « Implicit » indique que l'objet User Parameter est adressé de façon implicite par les services.

List of APs

Un API se compose des éléments de liste suivants:

API

Cet attribut définit le numéro du processus d'application.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

Real Identification

Cet attribut est constitué des attributs suivants:

List of Slots

Cet attribut définit les baies du processus d'application. Une baie (Slot) se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient le Real Identification Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 7FFF

Module Ident Number

Cet attribut définit l'identification de la baie.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFFFFF

NOTE 1 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Module Properties

Cet attribut est réservé pour usage futur.

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle le Real Identification Data Element appartient.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Submodule Ident Number

Cet attribut définit l'identification de la sous-baie.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

NOTE 2 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Submodule Properties

Cet attribut est constitué des attributs suivants.

Type

Cet attribut définit le type de la sous-baie.

Type d'attribut: Unsigned16

Valeurs permises: NO_IO, INPUT, OUTPUT, IO

Shared Input

Cet attribut définit les propriétés de la sous-baie. La valeur SHARED_INPUT doit être utilisée seulement si l'attribut «Submodule type» contient la valeur INPUT ou IO.

Type d'attribut: Unsigned8

Valeurs permises: SHARED_INPUT, NOT_SHARED_INPUT

Reduce Input Submodule Data Length

Cet attribut définit s'il est souhaité d'utiliser la longueur prévue des données du sous-module d'entrée pour la CR d'entrée ou de réduire la longueur des données à zéro.

Type d'attribut: Unsigned8

Valeurs permises: EXPECTED, ZERO

Reduce Output Submodule Data Length

Cet attribut définit s'il est souhaité d'utiliser la longueur prévue des données du sous-module de sortie pour la CR de sortie ou de réduire la longueur des données à zéro.

Type d'attribut: Unsigned8

Valeurs permises: EXPECTED, ZERO

Discard IOXS

Cet attribut définit s'il est souhaité de transférer l'IOXS prévu pour ce sous-module dans les CR correspondantes ou non.

Type d'attribut: Unsigned8

Valeurs permises: EXPECTED, ZERO

List of Data Descriptions

Cette liste d'attributs est constituée des attributs suivants:

Data Direction

L'attribut définit le sens entrée ou sortie.

Type d'attribut: Unsigned16

Valeurs permises: DIRECTION_INPUT, DIRECTION_OUTPUT

Submodule Data Length

L'attribut Submodule Data Length définit le nombre d'octets dans le Data Element sans compter l'IOPS.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 1 439

Length IOPS

L'attribut Length IOPS définit le nombre d'octets du statut de fournisseur.

Type d'attribut: Unsigned8

Valeurs permises: 1, (2 à 255 sont réservées)

Length IOCS

L'attribut Length IOCS définit le nombre d'octets du statut de consommateur.

Type d'attribut: Unsigned8

Valeurs permises: 1, (2 à 255 sont réservées)

Expected Identification

Cet attribut est constitué des attributs suivants:

List of ARs

Une AR se compose des éléments de liste suivants:

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

SYNC State

Cet attribut est le statut de synchronisation locale pour l'AR souhaitée. Il doit être présent seulement si la synchronisation est utilisée. Le comportement doit être selon 7.3.7.

Type d'attribut: Unsigned16

Valeurs permises: SYNCHRONIZED, SYNC_NOT_AVAILABLE

List of CRs

Une CR se compose des éléments de liste suivants:

CREP

Ce paramètre est l'identificateur local de la CR consommateur en multidiffusion souhaitée.

Multicast Provider State

Ce paramètre est le statut local de la CR consommateur en multidiffusion. Le comportement doit être selon 7.3.7.

Type d'attribut: Unsigned16

Valeurs permises: UP_AND_RUNNING,
MULTICAST_PROVIDER_NOT_AVAILABLE

List of APIs

Un API se compose des éléments de liste suivants:

API

Cet attribut définit le numéro du processus d'application.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

List of Slots

Une baie (Slot) se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient l'IO Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x7FFF

Module Ident Number

Cet attribut définit l'identification de la baie.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFFFFF

NOTE 3 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Module Properties

Cet attribut est réservé pour usage futur.

Module State

Cet attribut définit l'état du module et doit être utilisé pour le service "Connect" et le service "Application Ready".

Type d'attribut: Unsigned16

Valeurs permises: Doit être mis à des valeurs conformes au Tableau 169.

Tableau 169 – Module State

Valeur	Utilisation
NO_MODULE	La baie existe et le module n'est pas branché.
WRONG_MODULE	L'attribut Module Ident Number est erroné. Il convient d'éviter ce codage; utiliser « Substitute » pour le module et « Wrong (WR) » pour chaque sous-module.
PROPER_MODULE	La baie existe et le module est correct.
SUBSTITUTE	La baie existe et le module n'est pas le même que celui demandé - mais est compatible. L'appareil E/S était capable d'adapter sa propre décision. Il convient d'utiliser ce codage en conjonction avec « Wrong (WR) » pour chaque sous-module.

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient l'IO Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Submodule Ident Number

Cet attribut définit l'identification de la sous-baie.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

NOTE 4 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Submodule Properties

Cet attribut est constitué des attributs suivants.

Type

Cet attribut définit le type de la sous-baie.

Type d'attribut: Unsigned16

Valeurs permises: NO_IO, INPUT, OUTPUT, IO

Le Tableau 170 montre les dépendances qui doivent être appliquées conjointement au type de CR de l'attribut CR sélectionné.

Tableau 170 – Usage en rapport avec le type de CR

Type de sous-module	Usage avec le type de CR
NO_IO	INPUT CR, OUTPUT CR
INPUT	INPUT CR, OUTPUT CR, MULTICAST PROVIDER CR
OUTPUT	INPUT CR, OUTPUT CR, MULTICAST CONSUMER CR
IO	INPUT CR, OUTPUT CR, MULTICAST PROVIDER CR (partie Input seulement), MULTICAST CONSUMER CR (partie Output seulement)

Shared Input

Cet attribut définit les propriétés de la sous-baie. La valeur SHARED_INPUT doit être utilisée seulement si l'attribut «Submodule type» contient la valeur INPUT ou IO.

Type d'attribut: Unsigned8

Valeurs permises: SHARED_INPUT, NOT_SHARED_INPUT

Reduce Input Submodule Data Length

Cet attribut définit s'il est souhaité d'utiliser la longueur prévue des données du sous-module d'entrée pour la CR d'entrée ou de réduire la longueur des données à zéro.

Type d'attribut: Unsigned8

Valeurs permises: EXPECTED, ZERO

Reduce Output Submodule Data Length

Cet attribut définit s'il est souhaité d'utiliser la longueur prévue des données du sous-module de sortie pour la CR de sortie ou de réduire la longueur des données à zéro.

Type d'attribut: Unsigned8

Valeurs permises: EXPECTED, ZERO

Discard IOXS

Cet attribut définit s'il est souhaité de transférer l'IOXS prévu pour ce sous-module dans les CR correspondantes ou non.

Type d'attribut: Unsigned8

Valeurs permises: EXPECTED, ZERO

Submodule State

Cet attribut est constitué des attributs suivants:

Format Indicator

Cet attribut indique le format de Submodule State.

Valeurs permises: 0 (signification = V_1_0), 1 (signification > V_1_0)

Detail

Cet attribut contient la valeur de l'état du sous-module (Format Indicator = V_1_0).

Doit être mis à des valeurs conformes au Tableau 171.

Tableau 171 – Detail

Valeur	Signification
NO_SUBMODULE	La sous-baie existe et est vide.
WRONG_SUBMODULE	La sous-baie existe et un sous-module compatible est branché.
LOCKED_BY_IO_CONTROLLER	La sous-baie existe et est verrouillée par un autre appareil de commande E/S.
APPLICATION_READY_PENDING	Ce détail est délivré de manière statique ou dynamique. "Statique" signifie que le sous-module existe et un défaut de paramètre a été détecté ou l'application du sous-module n'est pas en fonctionnement. "Dynamique" signifie que le sous-module existe, mais l'exécution du paramétrage n'est pas encore prête. Dans ce cas, le sous-module émettra une alarme Return of Submodule, lorsqu'il sera prêt.
SUBSTITUTE	La sous-baie existe et le sous-module n'est pas le même que celui demandé – mais est compatible et l'appareil E/S était capable de s'adapter. Dans ce cas, l'appareil E/S doit s'adapter, par exemple à une nouvelle longueur d'entrée ou de sortie.
GOOD	La sous-baie existe et un sous-module prévu est branché.

NOTE 5 Le champ Submodule State est donné en réponse seulement s'il existe une différence entre les données de configuration prévue et réelle. Le Submodule State "GOOD" n'est pas défini, car pour de tels sous-modules, aucun statut n'est rapporté dans la réponse.

Add Info

Cet attribut contient des informations supplémentaires pour la relation entre applications (Format Indicator > V_1_0).

Pour une relation entre applications avec AR Type IOCAR ou IOCAR_SR, la valeur est toujours mise sur NO_ADD_INFO.

Pour une relation entre applications avec AR Type IOSAR, la valeur TAKEOVER_NOT_ALLOWED indique que le sous-module a été configuré avec AR Properties «Supervisor Takeover allowed» mis à «false» et, ainsi, le superviseur E/S n'a eu aucun droit de propriété pour le sous-module.

Valeurs permises: NO_ADD_INFO, TAKEOVER_NOT_ALLOWED

Qualified Info

Cet attribut indique si des informations de diagnostic qualifié sont présent dans au moins une des voies du sous-module (Format Indicator > V_1_0).

Qualifier Info est déduit de l'ASE Diagnosis; voir 7.3.4.4.5 pour plus de détails.

Valeurs permises: NO_QUALIFIED_INFO, QUALIFIED_INFO

NOTE 6 Si Shared Input contient la valeur SHARED_INPUT, ce champ contient toujours la valeur NO_QUALIFIED_INFO (voir 7.4.6.2.2.1 pour plus de détails).

Maintenance Required

Cet attribut indique si la maintenance requise est présente dans au moins une des voies du sous-module (Format Indicator > V_1_0).

Maintenance Required est déduite de l'ASE Diagnosis; voir 7.3.4.4.3 pour plus de détails.

Valeurs permises: NO_MAINTENANCE_REQUIRED, MAINTENANCE_REQUIRED

NOTE 7 Si Shared Input contient la valeur SHARED_INPUT, ce champ contient toujours la valeur NO_MAINTENANCE_REQUIRED (voir 7.4.6.2.2.1 pour plus de détails).

Maintenance Demanded

Cet attribut indique si la maintenance exigée est présente dans au moins une des voies du sous-module (Format Indicator > V_1_0).

Maintenance Demanded est déduite de l'ASE Diagnosis; voir 7.3.4.4.4 pour plus de détails.

Valeurs permises: NO_MAINTENANCE_DEMANDED, MAINTENANCE_DEMANDED

NOTE 8 Si Shared Input contient la valeur SHARED_INPUT, ce champ contient toujours la valeur NO_MAINTENANCE_DEMANDED (voir 7.4.6.2.2.1 pour plus de détails).

Diag Info

Cet attribut indique si des informations de diagnostic sont présent dans au moins une des voies du sous-module (Format Indicator > V_1_0).

Diag Info est déduit de l'ASE Diagnosis; voir 7.3.4.4.2 pour plus de détails.

Valeurs permises: NO_DIAG, DIAG

NOTE 9 Si Shared Input contient la valeur SHARED_INPUT, ce champ contient toujours la valeur NO_DIAG (voir 7.4.6.2.2.1 pour plus de détails).

AR Info

Cet attribut contient l'état de sous-module en relation avec l'AR établie (Format Indicator > V_1_0).

AR Info est déduit de Ownership du sous-module; voir le paragraphe 7.4.6.2.2 pour plus de détails.

Il doit être mis conformément au Tableau 172.

Tableau 172 – AR Info

Valeur	Signification
OWN	Cet AR est propriétaire du sous-module
APPLICATION_READY_PENDING	Cette AR est propriétaire du sous-module, mais le sous-module est bloqué par la vérification des paramètres en cours; émise de façon statique ou dynamique. "Statique" signifie que le sous-module existe et un défaut de paramètre a été détecté ou l'application du sous-module n'est pas en fonctionnement. "Dynamique" signifie que le sous-module existe, mais l'exécution du paramétrage n'est pas encore prête. Dans ce cas, le sous-module émettra une alarme Return of Submodule, lorsqu'il sera prêt.

Valeur	Signification
SUPERORDINATED_LOCKED	Cet AR n'est pas propriétaire du sous-module. Il est bloqué par des moyens superordonnés.
LOCKED_BY_IO_CONTROLLER	Cet AR n'est pas propriétaire du sous-module. Il est la propriété d'une autre IOAR
LOCKED_BY_IO_SUPERVISOR	Cet AR n'est pas propriétaire du sous-module. Il est la propriété d'une autre IOSAR

NOTE 10 Si Shared Input contient la valeur SHARED_INPUT, ce champ contient toujours la valeur OWN (voir 7.4.6.2.2.1 pour plus de détails).

Ident Info

Cet attribut contient l'état de sous-module en relation avec l'identification prévue et réelle pour une AR établie (Format Indicator > V_1_0).

Ident Info est déduit du sous-module Real; voir 7.4.6.2.1 pour plus de détails.

Il doit être mis conformément au Tableau 173.

Tableau 173 – Ident Info

Valeur	Signification	Utilisable conjointement au Module State "WRONG MODULE"
OK	Aucune différence	Non
SUBSTITUTE	Différent mais compatible	Non
WRONG	Incompatible	Oui
NO	Vide	Oui

NOTE 11 Le champ Ident Info contient la valeur OK, si le sous-module réel est celui attendu. La valeur SUBSTITUTE est renvoyée, si le sous-module réel est capable de se substituer à celui attendu. La valeur WRONG est renvoyée si un sous-module réel est branché, mais ne correspond pas à celui qui est attendu par rapport au type (ModuleID, SubmoduleID). La valeur NO est renvoyée si la sous-baie est vide, car aucun sous-module n'est branché. Si Shared Input contient la valeur SHARED_INPUT, ce champ contient toujours la valeur OK (voir 7.4.6.2.2.1 pour plus de détails).

I&M Data

Cet attribut est constitué des attributs suivants:

List of Slots

Cet attribut définit les baies du processus d'application. Une baie (Slot) se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient l'I&M Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 7FFF

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient l'I&M Data Element.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

I&M0 Data

Cet attribut est constitué des attributs suivants.

Vendor ID

Cet attribut définit l'ID du fabricant du sous-module. La valeur 0x0000 est utilisée au cas où un fabricant ne souhaiterait pas gérer un ID.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0xFFFF

Order ID

Cet attribut définit le numéro de commande complet ou au moins une partie pertinente qui permet une identification sans ambiguïté du sous-module dans le spectre de commandes du fabricant.

L'attribut est rempli comme "visible string" (chaîne visible). Les octets non utilisés doivent être mis à 0x20 (caractère espace).

Type d'attribut: VisibleString [20]

Serial Number

Cet attribut définit le numéro de série du sous-module. Le numéro de série est un numéro de production unique du fabricant d'appareil, et ce, même pour des appareils ayant la même édition du matériel, du logiciel ou du firmware.

L'attribut est rempli comme "visible string" (chaîne visible). Les octets non utilisés doivent être mis à 0x20 (caractère espace).

Type d'attribut: VisibleString [16]

Hardware Revision

Cet attribut définit l'édition du matériel du sous-module.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0xFFFF

Software Revision

Cet attribut définit l'édition du logiciel ou du firmware du sous-module. La structure prend en charge une différenciation grossière et détaillée qui peut être définie par le fabricant comme étant "Vx.y.z". Tout composant qui porte un logiciel doit présenter sa version de logiciel, et ce, même si l'utilisateur peut ne pas en être conscient. V255.255.255 indique la disponibilité d'informations spécifiques à un profil.

Cet attribut est constitué des attributs suivants:

Prefix

L'attribut de préfixe constitue la partie «Vx.y.z» de la révision de logiciel.

Type d'attribut: VisibleString [1]

Valeurs préférentielles:

"V" (= version sortie officiellement) ou "R" (= Révision)

"P" = Prototype

"U" = Under Test, c'est-à-dire "en essai" (essai de terrain)

"T" = Test Device (appareil d'essai)

Functional Enhancement

L'attribut Functional Enhancement (amélioration fonctionnelle) constitue la partie "Vx.y.z" de la révision de logiciel.

Type d'attribut: Unsigned8

Valeurs permises: 0 à 0xFF

Bug Fix

L'attribut Bug Fix (débugage) constitue la partie "Vx.y.z" de la révision de logiciel.

Type d'attribut: Unsigned8

Valeurs permises: 0 à 0xFF

Internal Change

L'attribut Minor Change (modification mineure) constitue la partie «Vx.y.z» de la révision de logiciel.

Type d'attribut: Unsigned8

Valeurs permises: 0 à 0xFF

Revision Counter

Cet attribut définit le compteur des révisions du sous-module.

Un changement du compteur des révisions marque un changement du matériel ou de ses paramètres. Au moment de la production, le compteur sera mis à 0. Cette valeur est réservée pour la première installation et le premier incrément. Ainsi, le compteur s'incrémente de 1... 65 535, repassant à 1 à la fin.

Le compteur est incrémenté à la suite d'un échange de sous-module ou sur tout accès en écriture de paramètres au sein du sous-module et son activation.

Pour éviter d'inutiles incréments du compteur, des critères plus tolérants et sophistiqués qui satisfont aux exigences minimales de la FDA (Food & Drug Administration) peuvent être mis en œuvre de façon spécifique à un fabricant (par exemple: ID de lot, serveur de paramètres locaux, etc.).

Le compteur des révisions dans le représentant de l'ensemble de l'appareil modulaire – habituellement affecté à la baie 0 – doit être incrémenté si un quelconque compteur de révisions du module est incrémenté.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0xFFFF

Profile ID

Cet attribut définit l'ID du profil d'application que le sous-module suit. Les sous-modules qui ne suivent aucun profil d'application PROFIBUS doivent utiliser 0xF600.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0xFFFF

Profile Specific Type

Si un sous-module suit un profil d'application particulier, cet attribut propose des informations relatives à l'utilisation de ses voies et/ou sous-appareils. Les informations relatives aux sous-modules/voies doivent être conformes aux définitions respectives du profil d'application.

Si un sous-module ne suit pas un profil d'application spécial, l'attribut indique le type de capteur ou d'actionneur.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0xFFFF

I&M Version

Cet attribut indique la version mise en œuvre des fonctions I&M.

L'octet 1 (MSB, c'est-à-dire de poids fort) contient le numéro de version majeure, par exemple le 1 dans version 1.0; l'octet 2 (LSB, c'est-à-dire de poids faible) contient le numéro de version mineure, par exemple le 0 version 1.0.

Type d'attribut: OctetString[2]

I&M Supported

Cet attribut indique la disponibilité d'autres enregistrements I&M. Le Bit 0 indique la disponibilité de l'I&M spécifique au Profil, le Bit 1 la disponibilité de l'I&M1, le Bit 2 la disponibilité de l'I&M2, ... et le Bit 15 la disponibilité de l'I&M15.

Type d'attribut: Bit Sequence V2

I&M1 Data

Cet attribut facultatif est constitué des attributs suivants:

Function

Cet attribut indique la fonction ou la tâche du sous-module. Il peut s'agir d'une étiquette symbolique normalisée ("AKZ") prise dans une liste ou de n'importe quel autre type d'étiquette défini par un outil de configuration. Les caractères non utilisés doivent être mis à 0x20 (caractère espace).

Type d'attribut: VisibleString [32]

Valeur par défaut: Rempli de 0x20 (caractère espace)

Location

Cet attribut indique l'emplacement du sous-module. Il peut s'agir d'une étiquette symbolique normalisée ("OKZ") prise dans une liste ou de n'importe quel autre type d'étiquette défini par un outil de configuration. Les caractères non utilisés doivent être mis à 0x20 (caractère espace).

Type d'attribut: VisibleString [22]

Valeur par défaut: Rempli de 0x20 (caractère espace)

I&M2 Data

Cet attribut facultatif est constitué de l'attribut suivant:

Installation Date

Cet attribut indique la date d'installation ou de mise en service d'un appareil ou d'un module. Le contenu est un format de date et d'heure fixe conforme à l'ISO 8601: AAAA-MM-JJ T HH:MM, où AAAA est l'année du calendrier grégorien habituel, MM est le mois de l'année entre 01 (janvier) et 12 (décembre), JJ est le jour du mois entre 01 et 31, HH est l'heure du jour entre 0 et 23 et MM est le nombre de minutes de l'heure entre 0 et 59.

L'heure et la date sont normalement séparées par un « T », mais pour cette apparence doivent être séparées par un caractère espace.

Par exemple, le quatrième jour du mois de février de l'année 1995 s'écrit en notation normalisée sous la forme:

1995-02-04 16:23

Les caractères non utilisés doivent être mis à 0x20 (caractère espace).

Type d'attribut: VisibleString [16]

Valeur par défaut: Rempli de 0x20 (caractère espace)

I&M3 Data

Cet attribut facultatif est constitué des attributs suivants:

Descriptor

Cet attribut permet de stocker n'importe quelles informations et annotations supplémentaires individuelles.

Les caractères non utilisés doivent être mis à 0x20 (caractère espace).

Type d'attribut: VisibleString [54]

Valeur par défaut: Rempli de 0x20 (caractère espace)

I&M4 Data

Cet attribut facultatif est constitué des attributs suivants:

Signature

Cet attribut permet à des outils de paramétrage de stocker un code de «sécurité» comme référence pour une session particulière de paramétrage et à des outils de piste d'audit de récupérer le code pour des vérifications d'intégrité. Conjointement à l'"asset identification" (identification de biens) constituée du Vendor ID, de l'Order ID et du Serial Number, l'attribut Signature procure une absence d'ambiguïté des données de session.

Les caractères non utilisés doivent être mis à 0x00.

Type d'attribut: OctetString[54]

Valeur par défaut: Renseignée par 0x00

I&M Filter Data

Cet attribut est constitué des attributs suivants:

I&M Carrier Data

Cet attribut contient tous les sous-modules qui sont porteurs de données d'identification et de maintenance (voir 7.1.4.5.7 pour la définition).

List of APs

Un API se compose des éléments de liste suivants:

API

Cet attribut définit le numéro du processus d'application.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

List of Slots

Cet attribut définit les baies du processus d'application. Une baie (Slot) se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient l'I&M.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 7FFF

Module Ident Number

Cet attribut définit l'identification de la baie.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFFFFF

NOTE 12 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient l'I&M.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Submodule Ident Number

Cet attribut définit l'identification de la sous-baie.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

NOTE 13 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Module Representative Data

Cet attribut facultatif indique les sous-modules qui agissent comme représentant du module et portent donc les données I&M du module. Il est constitué des attributs suivants:

NOTE 14 L'attribut indique au maximum un sous-module par module agissant comme représentant de module.

List of APs

Un API se compose des éléments de liste suivants:

API

Cet attribut définit le numéro du processus d'application.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

List of Slots

Cet attribut définit les baies du processus d'application. Une baie (Slot) se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient le représentant du module.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 7FFF

Module Ident Number

Cet attribut définit l'identification de la baie.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFFFFF

NOTE 15 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient le représentant du module.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Submodule Ident Number

Cet attribut définit l'identification de la sous-baie.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

NOTE 16 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Device Representative Data

Cet attribut contient un sous-module, qui est porteur d'informations d'identification et de maintenance (voir 7.1.4.5.7 pour la définition) et offre un accès à I&M1, I&M2, I&M3 et I&M4.

Il est constitué des attributs suivants:

NOTE 17 L'attribut indique au maximum un sous-module par module agissant comme représentant d'appareil.

List of APs

Un API se compose des éléments de liste suivants:

API

Cet attribut définit le numéro du processus d'application.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

List of Slots

Cet attribut définit les baies du processus d'application. Une baie (Slot) se compose des éléments de liste suivants:

Slot Number

Cet attribut définit le module auquel appartient le représentant de l'appareil.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 9FFF

Module Ident Number

Cet attribut définit l'identification de la baie.

Type d'attribut: Unsigned32

Valeurs permises: 1 à 0xFFFFFFFF

NOTE 18 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

List of Subslots

Cet attribut est constitué des attributs suivants.

Subslot Number

Cet attribut définit la sous-baie à laquelle appartient le représentant de l'appareil.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 0x9FFF

Submodule Ident Number

Cet attribut définit l'identification de la sous-baie.

Type d'attribut: Unsigned32

Valeurs permises: 0 à 0xFFFFFFFF

NOTE 19 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails

Autoconfiguration Data

Cet attribut facultatif permet d'établir une relation entre applications à un appareil E/S sans avoir besoin d'un système d'ingénierie. C'est pour cela que l'appareil E/S renvoie des informations concernant ses capacités de communication et énumère ses sous-modules qui sont en mesure d'être utilisés sans avoir besoin d'un paramétrage de démarrage.

Il est constitué des attributs suivants:

Communication Data

Cet attribut définit les capacités de communication. Il est constitué des attributs suivants:

Min Device Interval

Cet attribut indique l'intervalle minimal d'appareil pris en charge par l'appareil E/S en multiples de 31,25 µs.

Type d'attribut: Unsigned16

Valeurs permises: 0x1 à 0x1000

Base de temps: 31,25 µs

Communication Properties

Cet attribut indique les propriétés de communication de l'appareil E/S. Il est constitué des attributs suivants:

DFP

Cet attribut indique si l'appareil E/S a la capacité ou non de mise en paquet dynamique de trame. Si cet attribut est mis sur TRUE, l'appareil E/S doit prendre en charge RT_CLASS_3.

Type d'attribut: Boolean

RTC3

Cet attribut indique si l'appareil E/S prend en charge RT_CLASS_3.

Type d'attribut: Unsigned8

Valeurs permises:

- 0x00 (RT Class 3 non prise en charge),
- 0x01 (RT Class 3 supported, PD IR Frame Data needed),
- 0x02 (RT Class 3 supported, PD IR Frame Data not supported),
- 0x03 (RT Class 3 prise en charge, fonctionnement avec et sans prise en charge de PD IR Frame Data)

Send Clock Factors

Cet attribut contient les horloges d'envoi prises en charge par l'appareil E/S. L'attribut est interprété comme une série de bits. Chaque bit de la matrice correspond à un Send Clock Factor et doit être calculé comme suit:

$$N = (\text{Send Clock Factor} - 1) \text{ DIV } 8$$

$$M = (\text{Send Clock Factor} - 1) \text{ MOD } 8$$

où

N est le numéro de l'octet et

M est le numéro du bit de l'octet.

L'horloge d'envoi elle-même est calculée en multipliant le Send Clock Factor avec 31,25 µs. Un bit particulier doit être mis à TRUE, si le facteur de l'horloge d'envoi est pris en charge.

Type d'attribut: OctetString[16]

Configuration Data

Cet attribut contient les informations de configuration automatique liées à la configuration. Il contient les sous-modules de l'appareil E/S utilisable pour un fonctionnement sans aucun paramétrage (c'est-à-dire que le sous-module est utilisable avec une valeur par défaut appropriée). Si un sous-module n'est pas utilisable sans paramétrage, il ne doit pas faire partie de cette liste.

Les données de configuration se composent des attributs suivants:

List of Offered Submodules

Cet attribut contient les Configuration Data Elements (éléments de données de configuration). Un élément de liste décrit un sous-module particulier utilisable sans paramétrage et se compose des attributs suivants:

API

Cet attribut contient l'indice de profil d'application du profil d'application auquel le sous-module appartient.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000000 à 0xFFFFFFFF

Slot Number

Cet attribut contient le numéro de baie contenant le sous-module.

Type d'attribut: Unsigned16

Valeurs permises: 0x0000 à 0x7FFF

Subslot Number

Cet attribut contient le numéro de sous-baie contenant le sous-module.

Type d'attribut: Unsigned16

Valeurs permises: 0x0000 à 0x7FFF

Module Ident Number

Cet attribut contient l'identification du module de la baie contenant le sous-module.

Type d'attribut: Unsigned32

Valeurs permises: 0x1 à 0xFFFFFFFF

NOTE 19 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Submodule Ident Number

Cet attribut contient l'identification du sous-module.

Type d'attribut: Unsigned32

Valeurs permises: 0x1 à 0xFFFFFFFF

NOTE 21 La valeur d'attribut est spécifique à un fabricant. Voir 7.1.4.5.5 pour plus de détails.

Length Input Data

L'attribut contient le nombre d'octets de l'élément de données d'entrée du sous-module sans IOPS.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 1 439

Length Output Data

L'attribut contient le nombre d'octets de l'élément de données de sortie du sous-module sans IOPS.

Type d'attribut: Unsigned16

Valeurs permises: 0 à 1 439

Isochronous Data

Cet attribut facultatif contient les informations de configuration automatique liées au sous-module isochrone utilisable. Il contient les sous-modules de l'appareil E/S utilisable pour un fonctionnement isochrone sans aucun paramétrage (c'est-à-dire que le sous-module est utilisable avec une valeur par défaut appropriée), en plus des paramètres isochrones. Si un sous-module n'est pas utilisable de façon isochrone sans paramétrage, il ne doit pas faire partie de cette liste.

Chaque sous-module fourni dans les données isochrones doit également faire partie des données de configuration. Ainsi le numéro d'identification du module, du sous-module et la longueur de données d'entrée et de sortie sont utilisés à partir de l'entrée appropriée dans les données de configuration.

Les données isochrones se composent des attributs suivants:

List of Offered Submodules

Cet attribut contient les Isochronous Data Elements (éléments de données isochrones). Un élément de liste décrit un sous-module particulier utilisable sans paramétrage et se compose des attributs suivants:

API

Cet attribut contient l'indice de profil d'application du profil d'application auquel le sous-module appartient.

Type d'attribut: Unsigned32

Valeurs permises: 0x00000000 à 0xFFFFFFFF

Slot Number

Cet attribut contient le numéro de baie contenant le sous-module.

Type d'attribut: Unsigned16

Valeurs permises: 0x0000 à 0x7FFF

Subslot Number

Cet attribut contient le numéro de sous-baie contenant le sous-module.

Type d'attribut: Unsigned16

Valeurs permises: 0x0000 à 0x7FFF

Time IO Base (T_IO_Base)

Cet attribut détermine la granularité pour le réglage du délai d'entrée et de sortie.

Le temps pour T_IO_Base est calculé par $T\_IO\_Base * 1 \text{ ns}$.

Base de temps: 1 ns

Type d'attribut: Unsigned32

Valeurs permises: 1 à 32 000 000

Time IO Input (T_IO_Input)

Cet attribut contient le facteur pour le délai de rafraîchissement d'entrée possible. Il peut s'agir du temps le plus petit possible pour copier ou prétraiter les données d'entrée.

Le temps pour T_IO_Input est calculée par $T\_IO\_Input * T\_IO\_Base * 1 \text{ ns}$.

Base de temps: $T\_IO\_Base * 1 \text{ ns}$

Type d'attribut: Unsigned32

Valeurs permises: 1 à 32 000 000

Time IO Output (T_IO_Output)

Cet attribut contient le facteur pour le délai de rafraîchissement de sortie possible. Il peut s'agir du temps le plus petit possible pour copier ou prétraiter les données de sortie.

Le temps pour T_IO_Output est calculée par $T\_IO\_Output * T\_IO\_Base * 1 \text{ ns}$.

Base de temps: $T\_IO\_Base * 1 \text{ ns}$

Type d'attribut: Unsigned32

Valeurs permises: 1 à 32 000 000

Fast Startup Data

Cet attribut facultatif permet d'ajuster le comportement de démarrage rapide par rapport à chaque relation entre applications

Il est constitué des attributs suivants:

List of ARs

Une AR se compose des éléments de liste suivants:

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Expected Fast Startup Data

Cet attribut contient le Fast Startup Parameter Data prévu pour cette AR. Il se compose des attributs suivants:

FS Parameter Block

Cet attribut facultatif contient le Fast Startup Parameter Data pour l'AR. Il se compose des attributs suivants:

FS Parameter Mode

Cet attribut contient le mode de paramètre actuellement actif de l'AR.

Type d'attribut: Unsigned32

Valeurs permises: ON, OFF

FS Parameter UUID

Cet attribut contient l'UUID donné par un système d'ingénierie pour indiquer les jeux de paramètres inchangés.

Type d'attribut: UUID

Manufacturer Specific Fast Startup Data

Cet attribut facultatif doit contenir les données de démarrage rapide spécifiques à un fabricant

Type d'attribut: OctetString

GSD Data

Cet attribut facultatif contient le GSD de cet appareil E/S.

Type d'attribut: OctetString

7.3.6.2.3 Invocation de l'objet Context

Chaque appareil E/S doit invoquer un objet Context d'IO Device.

Une demande de service de contexte (à savoir Connect, Prm Begin...) doit être traitée en moins de 300 s; autrement, le client du service peut supposer que la demande de service a expiré

7.3.6.2.4 Liaison des services de Context

Les services de l'ASE Context sont liés à aux services de niveau inférieur suivants (voir Tableau 174).

Tableau 174 – Liaison des services Context

Service	Liaison	Usage
Connect	lié à RPC::Connect	utilisé par l'appareil de commande E/S ou le superviseur E/S
Read	lié à Record Data::Read	utilisé par l'appareil de commande E/S ou le superviseur E/S

Service	Liaison	Usage
Write	lié à Record Data::Write	utilisé par l'appareil de commande E/S ou le superviseur E/S
Commande	lié à RPC::Control	utilisé par l'appareil de commande E/S ou le superviseur E/S ou un appareil E/S
Release	lié à RPC::Release	utilisé par l'appareil de commande E/S ou le superviseur E/S

7.3.6.3 Spécification de services de Context

7.3.6.3.1 Connect

7.3.6.3.1.1 Généralités

Deux types de service Connect sont pris en charge:

- Le service Connect (voir 7.3.6.3.1.2) pour établir une relation entre applications pour l'utilisation d'un appareil de commande E/S ou d'un superviseur E/S avec fonctions de statut et de commande.
- Le service Connect Device Access (voir 7.3.6.3.1.3) pour établir une relation entre applications pour l'utilisation d'un superviseur E/S sans fonctions de statut et de commande.

7.3.6.3.1.2 Connect

Le service Connect est utilisé pour établir une IO AR ou une Supervisor AR. Le processus d'application doit vérifier le paramètre de service de la demande/indication avec le paramètre local dans ses objets ASE (attributs Real Identification). En cas de discordance de paramètre de service (par exemple: des sous-modules sont déjà utilisés par d'autres IO AR) avec les attributs d'objets, le processus d'application doit répondre par les modules discordants dans la réponse. Le client peut entreprendre les actions appropriées. Si la phase d'établissement avait réussi, les paramètres de service sont stockés comme valeurs d'attribut dans le groupe d'attributs Expected Identification de l'objet Context d'IO Device.

Le Tableau 175 montre les paramètres du service.

NOTE 1 Les paramètres de service IP Address et Object UUID sont nécessaires pour adresser l'appareil de destination. Le modèle sous-jacent suppose que ces informations sont fournies par les informations d'AR, qui doivent être chargées auparavant pour le compte de l'application client. Du côté serveur, ces informations sont également gérées à l'intérieur de la couche application et n'apparaissent pas comme un paramètre de service explicite. Une mise en œuvre locale peut passer ce paramètre de service au processus d'application. Le même modèle s'applique aux services Read sur l'AR implicite.

L'application du serveur recevant la primitive "indication" du service Connect doit vérifier le paramètre de service. Notamment, les paramètres Module Ident Number, Submodule Ident Number, Submodule Data Length, IOCS Length et IOPS Length doivent être vérifiés seulement par l'application utilisateur.

Les ressources pour une relation entre applications sont allouées à Connect de la relation entre applications. Cela inclut les ressources pour la relation entre applications elle-même et les ressources des IOCR. Les ressources sont libérées lorsque la relation application est terminée.

Pour la première relation entre applications ayant un ARType IOCARSR et appartenant à un nouveau AR Set (comme indiqué par l'ARUUID; pour plus de détails, voir CEI 61158-6-10), les ressources de l'AR Set entier sont allouées; à savoir que pour toutes les AR de l'AR Set entier. Les ressources d'une relation entre applications appartenant au même AR Set établi à un moment ultérieur sont attribuées à partir de cette mémoire allouée. Les ressources de l'AR Set sont libérées lorsque la relation entre applications de l'AR Set est terminée.

Si la relation entre applications ne peut pas être établie en raison de la rareté des ressources, le Connect est rejeté avec une erreur «out of AR/ARset resources».

Si une relation entre applications doit contenir les sous-modules PDev qui sont déjà détenus par une autre relation entre applications, le Connect est rejeté avec une erreur «PDev already owned».

Les facteurs d'horloge d'envoi de toutes les relations entre applications établies en parallèle à un appareil E/S particulier IO doivent être identiques. Ainsi, le facteur d'horloge d'envoi de l'appareil E/S est déterminé par la première relation entre applications établie. Le Connect d'une relation entre applications réussie ayant un facteur d'horloge d'envoi différent est rejeté avec une erreur pour le bloc / paramètre correspondant.

Les IRDataUUID de toutes les relations entre applications utilisant RT_CLASS_3 établies parallèlement à un appareil E/S particulier doivent être identiques. L'IRDataUUID de l'appareil E/S est déterminé par la première relation entre applications établie avec RT_CLASS_3. Le Connect d'une relation entre applications réussie utilisant une RT_CLASS_3 ayant un IRDataUUID différent est rejeté avec une erreur pour le bloc / paramètre correspondant.

Tableau 175 – Connect

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
AR Parameter Block	M	M(=)		
AR type	M	M(=)		
AR UUID	M	M(=)		
Session Key	M	M(=)		
CM Initiator Mac Add	M	M(=)		
CM Initiator Object UUID	M	M(=)		
AR Properties	M	M(=)		
CM Initiator Activity Timeout Factor	M	M(=)		
Initiator UDP RT Port	M	M(=)		
CM Initiator Station Name	M	M(=)		
List of IO CR Parameter Blocks	M	M(=)		
IO CR type	M	M(=)		
IO CR Reference	M	M(=)		
LT Field	M	M(=)		
IO CR Properties	M	M(=)		
C_SDU Length	M	M(=)		
Frame ID	M	M(=)		
Send Clock Factor	M	M(=)		
Reduction Ratio	M	M(=)		
Phase	M	M(=)		
Sequence	M	M(=)		
Frame Send Offset	M	M(=)		
Data Hold Factor	M	M(=)		
IO CR Tag Header	M	M(=)		
IO CR Multicast MAC Add	M	M(=)		
List Of APIs	M	M(=)		
API	M	M(=)		

Nom de paramètre	req	ind	rsp	cnf
List of Related IO Data Objects	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
IO Data Object Frame Offset	M	M(=)		
List of Related IOCS	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
IOCS Frame Offset	M	M(=)		
Alarm CR Parameter Block	M	M(=)		
Alarm CR type	M	M(=)		
LT Field	M	M(=)		
Alarm CR Properties	M	M(=)		
RTA Timeout Factor	M	M(=)		
RTA Retries	M	M(=)		
Local Alarm Reference	M	M(=)		
Max Alarm Data Length	M	M(=)		
Alarm CR Tag Header High	M	M(=)		
Alarm CR Tag Header Low	M	M(=)		
List of Expected Submodule Blocks	U	U(=)		
List of Related APIs	M	M(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Module Ident Number	M	M(=)		
Module Properties	M	M(=)		
List of Submodules	M	M(=)		
Subslot Number	M	M(=)		
Submodule Ident Number	M	M(=)		
Submodule Properties	M	M(=)		
List of Data Descriptions	M	M(=)		
Data Description	M	M(=)		
Submodule Data Length	M	M(=)		
Length IOPS	M	M(=)		
Length IOCS	M	M(=)		
Parameter Server Block	U	U(=)		
Parameter Server Object UUID	M	M(=)		
Parameter Server Properties	M	M(=)		
CM Initiator Activity Timeout Factor	M	M(=)		
Parameter Server Station Name	M	M(=)		
List of Multicast CR Blocks	U	U(=)		
IO CR Reference	M	M(=)		
Address Resolution Properties	M	M(=)		
MCI Timeout Factor	M	M(=)		
Provider Station Name	M	M(=)		
AR RPC Block	U	U(=)		
Initiator RPC Server Port	M	M(=)		

Nom de paramètre	req	ind	rsp	cnf
IR Info Block	U	U(=)		
IR Data UUID	M	M(=)		
Number of IO CRs	M	M(=)		
List of Sub Frame Data	U	U(=)		
IO CR Reference	M	M(=)		
Sub Frame Offset	M	M(=)		
Sub Frame Data	M	M(=)		
SR Info Block	U	U(=)		
Redundancy Data Hold Factor	M	M(=)		
SR Properties	M	M(=)		
Input Valid on Backup AR	M	M(=)		
Activate Redundancy Alarm	M	M(=)		
AR Server Block	U	U(=)		
CM Initiator Station Name	M	M(=)		
List of AR Vendor Block Requests	U	U(=)		
AP Structure Identifier	M	M(=)		
API	M	M(=)		
Data Request	M	M(=)		
AR FSU Block	U	U(=)		
FS Parameter Block	U	U(=)		
FS Parameter Mode	M	M(=)		
FS Parameter UUID	M	M(=)		
Fast StartUp Block	U	U(=)		
Data	M	M(=)		
Result(✓)			S	S(=)
AREP			M	M(=)
AR Response Block			M	M(=)
AR type			M	M(=)
AR UUID			M	M(=)
Session Key			M	M(=)
CM Responder MAC Add			M	M(=)
Responder UDP RT Port			M	M(=)
List of IO CR Response Blocks			M	M(=)
IO CR type			M	M(=)
IO CR Reference			M	M(=)
Frame ID			M	M(=)
Alarm CR Response Block			M	M(=)
Alarm CR type			M	M(=)
Local Alarm Reference			M	M(=)
Max Alarm Data Length			M	M(=)
Module Diff Block			C	C(=)
List of APIs			M	M(=)
API			M	M(=)
List of Modules			M	M(=)

Nom de paramètre	req	ind	rsp	cnf
Slot Number			M	M(=)
Module Ident Number			M	M(=)
Module State			M	M(=)
List of Submodules			M	M(=)
Subslot Number			M	M(=)
Submodule Ident Number			M	M(=)
Submodule State			M	M(=)
AR RPC Block			C	C(=)
Responder RPC Server Port			M	M(=)
List of AR Vendor Block Responses			C	C(=)
AP Structure Identifier			M	M(=)
API			M	M(=)
Data Response			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

AR Parameter Block

Le paramètre contient les attributs suivants:

AR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

AR UUID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE 2 L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

Type d'attribut: Unsigned16

CM Initiator MAC Add

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

IO Initiator Object UUID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

AR Properties

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

CM Initiator Activity Timeout Factor

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Initiator UDP RT Port

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

CM Initiator Station Name

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

List of IO CR Parameter Blocks

Cette liste de paramètres contient les attributs suivants:

IO CR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

IO CR Reference

Ce paramètre contient la valeur pour identifier la CR dans ce service.

NOTE 3 Il est utilisé dans le contexte du service Connect pour relier les éléments IO Data à ses CR. Dans le contexte local de l'appareil E/S, il est transféré dans le CREP.

LT Field

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

IO CR Properties

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

C_SDU Length

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Frame ID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Send Clock Factor

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Reduction Ratio

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Phase

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Séquence

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Frame Send Offset

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Data Hold Factor

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

IO CR Tag Header

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

IO CR Multicast MAC Add

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

List Of APIs

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

API

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

List of Related IO Data Objects

La liste de paramètres doit toujours être présente. Elle contient les baies et les sous-baies dans lesquelles l'accès en lecture et en écriture est permis. Le processus d'application du serveur doit répondre par une réponse négative à des services Write adressés à des objets situés au-delà de cet espace adresse négocié. L'accès en écriture est toujours autorisé aux autres objets. Le paramètre suivant doit être vérifié avec les attributs correspondants de l'objet IO Data et le résultat en cas de réponse positive doit être stocké comme valeurs d'attributs de l'objet Context.

Il convient que l'utilisateur du service ne spécifie pas le même objet IO Data à plusieurs reprises au sein d'une IOCR, car cela peut ne pas être pris en charge par l'appareil E/S.

Le paramètre est constitué des sous-paramètres suivants:

Slot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Subslot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

IO Data Object Frame Offset

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

List of Related IOCS

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Slot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Subslot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

IOCS Frame Offset

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Alarm CR Parameter Block

Ce paramètre est constitué des attributs suivants:

Alarm CR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

LT Field

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Alarm CR Properties

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

RTA Timeout Factor

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

RTA Retries

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Local Alarm Reference

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Max Alarm Data Length

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Alarm CR Tag Header High

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Alarm CR Tag Header Low

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

List of Expected Submodule Blocks

Cette liste de paramètres est constituée des paramètres suivants:

NOTE 4 La liste inclut seulement des sous-modules «réels», mais n'inclut pas de baies «vides».

List Of Related APIs

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

API

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Slot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Module Ident Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Module Properties

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

List of Submodules

Cette liste est constituée des attributs suivants:

Subslot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Submodule Ident Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Submodule Properties

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

List of Data Descriptions

Cette liste est constituée des attributs suivants:

Data Description

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Submodule Data Length

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Length IOPS

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Length IOCS

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE Context.

Parameter Server Block

Ce paramètre est constitué des attributs suivants:

Parameter Server Object UUID

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Parameter Server Properties

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

CM Initiator Activity Timeout Factor

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Parameter Server Station Name

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

List of Multicast CR Block

Ce paramètre est constitué des attributs suivants:

IO CR Reference

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Address Resolution Properties

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

MCI Timeout Factor

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Provider Station Name

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

AR RPC Block

Ce paramètre facultatif doit être utilisé par l'appareil E/S pour adresser le port Initiator RPC Server Port en lieu et place du port de l'appareil de mapping de points d'extrémité. Il est constitué des attributs suivants:

Initiator RPC Server Port

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

IR Info Block

Ce paramètre facultatif doit être utilisé par l'appareil E/S pour déterminer si la relation entre applications fait référence aux données IR, ce qui fait partie du paramétrage PDev persistant. Le paramètre ne doit être utilisé que lorsque RT_CLASS_3 est utilisé avec le mode de démarrage avancé. Le paramètre permet à plusieurs appareils de commande E/S de partager l'appareil E/S en utilisant RT_CLASS_3. Un seul appareil de commande E/S peut paramétrer les données IR au PDev, tous les autres appareils de commande E/S se réfèrent aux données IR configurées par cet appareil de commande E/S via IR Info Block. Il est constitué des attributs suivants.

IR Data UUID

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Number of IO CRs

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

List of DFP IO CRs

Ce paramètre est constitué des attributs suivants:

IO CR Reference

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE CRL.

Sub Frame Offset

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Sub Frame Data

Ce paramètre est constitué des attributs suivants:

Sub Frame Position

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Sub Frame Data Length

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

SR Info Block

Cet attribut facultatif définit les paramètres de redondance du système relatifs à l'AR désirée. Cet attribut n'est présent que si le mode de démarrage de l'AR est avancé. Il est constitué des attributs suivants:

Redundancy Data Hold Factor

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

SR Properties

Cet attribut est utilisé pour définir le comportement de la redondance du système. Il est constitué des attributs suivants:

Input Valid on Backup AR

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Activate Redundancy Alarm

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

AR Server Block

Cet attribut facultatif n'est présent que si le mode de démarrage de l'AR est avancé. Il permet de vérifier, si l'appareil répondant au service de connexion est celui prévu, grâce à son nom. Il est constitué des attributs suivants:

CM Responder Station Name

Ce paramètre contient le nom du répondeur CM.

NOTE 5 La motivation pour ce bloc est la phase entre un DCP::Identify(name) et la connexion suivante à l'adresse IP de l'appareil. Si entre temps le nom de l'appareil était remplacé par, par exemple, DCP::Set, la connexion ne détecterait pas cette situation. Avec ce bloc, l'appareil de commande E/S est en mesure de vérifier si le nom == nom de la station répondeur CM. Dans le cas contraire, l'appareil de commande E/S annulerait alors la connexion.

List of AR Vendor Block Requests

Cet attribut facultatif permet de transférer les paramètres normatifs, spécifiques au fournisseur ou au profil pour l'AR désirée. Cet attribut n'est présent que si le mode de démarrage de l'AR est avancé.

Toute combinaison possible d'API et d'APStructureIdentifier peut exister uniquement une fois dans cette liste.

Un appareil E/S ne doit pas s'attendre à un ordonnancement dans cette liste.

Il est constitué des attributs suivants:

AP Structure Identifier

T Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

API

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Data Request

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

AR FSU Block

Cet attribut facultatif permet de transférer les paramètres spécifiques au démarrage rapide pour l'AR désirée. Cet attribut n'est présent que si le mode de démarrage de l'AR est avancé. Il est constitué des attributs suivants:

FS Parameter Block

Ce paramètre facultatif est constitué des éléments suivants:

FS Parameter Mode

Ce paramètre est constitué des éléments suivants:

Mode

Ce mode est utilisé pour établir le mode de démarrage rapide de l'application.

Type d'attribut: Unsigned8

Valeurs permises: OFF (0x00), ON (0x01)

Valeur par défaut: OFF (hors service)

FS Parameter UUID

Ce paramètre contient un UUID indiquant la génération de la configuration de données d'enregistrement pour l'appareil. L'UUID est modifié en recherchant chaque fois que quelque chose change dans les données d'enregistrement, lorsque des enregistrements sont ajoutés ou retirés de la configuration.

Type d'attribut: UUID

Fast Startup Block

Ce paramètre facultatif est constitué des éléments suivants:

Data

Ce paramètre conserve les données.

Type d'attribut: OctetString

Result (+)

Ce paramètre indique que la demande de service a réussi. Il doit seulement contenir un paramètre IO Data Response Block pour des entrées qui diffèrent des valeurs de paramètre demandées.

AR Response Block

Le paramètre sélectionnable contient les attributs suivants:

AR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

AR UUID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE 6 L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

Type d'attribut: Unsigned16

CM Responder MAC Add

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Responder UDP RT Port

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

List of IO CR Response Blocks

Cette liste de paramètres contient les attributs suivants:

IO CR Type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

IO CR Reference

Ce paramètre contient la valeur miroitée issue de la demande.

Frame ID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Alarm CR Response Block

Ce paramètre est constitué des attributs suivants:

Alarm CR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Local Alarm Reference

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Max Alarm Data Length

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe CRL.

Module Diff Block

Le Module Diff Block contient les sous-modules qui diffèrent de ceux qui ont été demandés; à savoir les sous-modules qui ne correspondent pas à la configuration prévue (erronés, manquants ou remplacés) et les sous-modules qui ne font pas partie de la relation entre applications (superordonnés verrouillés, verrouillés par un autre appareil de commande E/S ou verrouillés par le superviseur) ainsi que les sous-modules comportant des données diagnostic et/ou de maintenance.

Le paramètre est constitué des sous-paramètres suivants:

List of APIs

Ce paramètre contient les API qui diffèrent de la configuration demandée.

API

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

List of Modules

Ce paramètre contient les modules qui diffèrent de la configuration demandée.

Slot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Module Ident Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Module State

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

List of Submodules

Cette liste de paramètres est constituée des paramètres suivants:

Subslot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Submodule Ident Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Submodule State

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

AR RPC Block

Ce paramètre facultatif doit être utilisé par l'appareil E/S pour adresser le port Responder RPC Server Port en lieu et place du port de l'appareil de mapping de points d'extrémité. Il est constitué des attributs suivants:

Responder RPC Server Port

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

NOTE 7 Il est nécessaire d'interroger l'appareil de mise en correspondance de points d'extrémité pour obtenir ce port afin d'éviter les problèmes de pare-feu.

List of AR Vendor Block Response

Cet attribut facultatif permet de transférer les paramètres de réponse normatifs, spécifiques au fournisseur ou au profil pour l'AR désirée. Cet attribut n'est présent que si le mode de démarrage de l'AR est avancé et que si une demande AR Vendor Block a été utilisée.

Pour chaque demande AR Vendor Block, cette liste doit comporter une réponse Vendor Block dans le même ordre que dans la demande. La réponse ne doit pas comporter de blocs supplémentaires.

Il est constitué des attributs suivants:

AP Structure Identifier

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

API

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Data Response

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE AR.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Decode

Ce paramètre doit avoir la valeur PNIO.

Type: Unsigned8

Error code 1

L'erreur de code (Error code) 1 prend l'une des valeurs suivantes:
FAULTY_CONNECT_BLOCK.

Type: Unsigned16

Error code 2

Le paramètre Error Code 2 est spécifique à un utilisateur.

Type: Unsigned8

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 8 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 9 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

7.3.6.3.1.3 Connect Device Access

Le service Connect est utilisé pour établir une Supervisor AR pour l'accès aux appareils. Le processus d'application doit vérifier le paramètre de service de la demande/indication avec le paramètre local dans ses objets ASE (attributs Real Identification). Le client peut entreprendre les actions appropriées. Si la phase d'établissement avait réussi, les paramètres de service sont stockés comme valeurs d'attribut dans le groupe d'attributs Expected Identification de l'objet Context d'IO Device.

Le Tableau 176 montre les paramètres du service.

NOTE 1 Les paramètres de service IP Address et Object UUID sont nécessaires pour adresser l'appareil de destination. Le modèle sous-jacent suppose que ces informations sont fournies par les informations d'AR, qui doivent être chargées auparavant pour le compte de l'application client. Du côté serveur, ces informations sont également gérées à l'intérieur de la couche application et n'apparaissent pas comme un paramètre de service explicite. Une mise en œuvre locale peut passer ce paramètre de service au processus d'application. Le même modèle s'applique aux services Read sur l'AR implicite.

L'application du serveur recevant la primitive "indication" du service Connect doit vérifier le paramètre de service.

Tableau 176 – Connect Device Access

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
AR Parameter Block	M	M(=)		
AR type	M	M(=)		
AR UUID	M	M(=)		

Nom de paramètre	req	ind	rsp	cnf
Session Key	M	M(=)		
CM Initiator MAC Add	M	M(=)		
CM Initiator Object UUID	M	M(=)		
AR Properties	M	M(=)		
CM Initiator Activity Timeout Factor	M	M(=)		
Initiator UDP RT Port	M	M(=)		
CM Initiator Station Name	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
AR Response Block			M	M(=)
AR type			M	M(=)
AR UUID			M	M(=)
Session Key			M	M(=)
CM Responder MAC Add			M	M(=)
Responder UDP RT Port			M	M(=)
Result (-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

AR Parameter Block

Le paramètre contient les attributs suivants:

AR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

AR UUID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE 2 L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

Type d'attribut: Unsigned16

CM Initiator MAC Add

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

IO Initiator Object UUID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

AR Properties

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

CM Initiator Activity Timeout Factor

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Initiator UDP RT Port

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

CM Initiator Station Name

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Result (+)

Ce paramètre indique que la demande de service a réussi. Il doit seulement contenir un paramètre IO Data Response Block pour des entrées qui diffèrent des valeurs de paramètre demandées.

AR Response Block

Le paramètre sélectionnable contient les attributs suivants:

AR type

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

AR UUID

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE 3 L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

Type d'attribut: Unsigned16

CM Responder MAC Add

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Responder UDP RT Port

Ce paramètre contient la valeur pour l'attribut équivalent de la spécification de la classe ARL.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Decode

Ce paramètre doit avoir la valeur PNIO.

Type: Unsigned8

Error code 1

L'erreur de code (Error code) 1 prend l'une des valeurs suivantes:
FAULTY_CONNECT_BLOCK.

Type: Unsigned16

Error code 2

Le paramètre Error Code 2 est spécifique à un utilisateur.

Type: Unsigned8

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 4 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 5 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

7.3.6.3.2 Release

Ce service doit être utilisé par un appareil de commande E/S ou un superviseur E/S pour déconnecter une IO AR ou une Supervisor AR.

Le Tableau 177 montre les paramètres du service.

Tableau 177 – Release

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Session Key	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Session Key			M	M(=)
Result (-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE 1 L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Result(-)

Ce paramètre indique que la demande de service a échoué.

Error Decode

Ce paramètre doit contenir la valeur PNIO.

Type: Unsigned8

Error code 1

L'erreur de code (Error code) 1 prend l'une des valeurs suivantes:
FAULTY_RELEASE_BLOCK

Type: Unsigned16

Error code 2

Le paramètre Error Code 2 est spécifique à un utilisateur.

Type: Unsigned8

Add Data 1

Le paramètre Add Data est spécifique à un API (profil). La valeur 0 doit être émise si aucune donnée complémentaire 1 n'est définie.

Type: Unsigned 16

NOTE 2 Add Data 1 peut être utilisé par des spécifications de profils pour émettre des messages d'erreur spécifiques.

Add Data 2

Le paramètre Add Data est spécifique à un utilisateur. La valeur 0 doit être émise si aucune donnée complémentaire 2 n'est définie.

Type: Unsigned 16

NOTE 3 Add Data 2 peut être utilisé par des fournisseurs d'appareils pour émettre des messages d'erreur spécifiques.

7.3.6.3.3 Abort

Ce service doit être utilisé par un appareil E/S pour fermer une IO AR ou une Supervisor AR en cas d'erreur sérieuse lorsque le fonctionnement n'est plus possible. Le Tableau 178 montre les paramètres du service.

Tableau 178 – Abort

Nom de paramètre	req
Argument	M
AREP	M

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

7.3.6.3.4 Local AR Abort

Ce service doit être utilisé par un appareil E/S ou un superviseur E/S pour indiquer l'abandon local d'une AR existante. Le Tableau 179 montre les paramètres du service.

Tableau 179 – Local AR Abort

Nom de paramètre	req	ind	cnf
Argument	M(=)	M(=)	M(=)
AREP	M(=)	M(=)	M(=)
Session Key	M(=)	M(=)	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

Type d'attribut: Unsigned16

7.3.6.3.5 Prm Begin

Ce service permet une paramétrisation acquittée dans le contexte d'une AR établie.

Le service Prm Begin est orienté "un à un" et sur une relation entre applications indiquée par le paramètre AREP. L'émission du service Prm Begin est autorisée à tout moment après la notification IN-DATA pour l'AR désirée (après l'Application Ready.rsp). Le service Prm Begin permet de lancer une séquence de reparamétrisation acquittée pour les sous-modules indiqués par la List of Submodules.

Tous les sous-modules notés sont paramétrés au sein de la séquence de paramétrisation; pendant la séquence de paramétrisation, l'appareil de commande E/S livre un jeu cohérent de paramètres en émettant les services IODWrite.req ou autrement IODWriteMultiple.req. L'appareil de commande E/S paracheve la livraison de paramètre en émettant le service Prm End. L'absence de paramètres pour un sous-module devant être paramétré signifie que le sous-module doit utiliser les paramètres par défaut.

La séquence de paramétrisation est terminée par l'émission d'Application Ready par l'appareil E/S vers l'appareil de commande E/S, réduite à nouveau aux sous-modules contenus dans la séquence de paramétrisation et contenant un ModuleDiffBlock orienté vers les sous-modules indiqués dans la demande. Après ce service Application Ready, les sous-modules sont à nouveau utilisables pour l'appareil de commande E/S.

Pour les paramètres IOxS des sous-modules concernés, les règles relatives au paramétrage du démarrage "normale" s'appliquent: Ils sont mis à "Good" par l'appareil E/S dès que le paramétrage du sous-module approprié est terminé et le sous-module est en mesure de livrer ou de recevoir des données en fonction des nouveaux paramètres.

La liste des sous-modules doit posséder les mêmes sous-modules, ou un sous-ensemble de ces sous-modules énumérés, que la liste de sous-modules prévus au sein du service Connect pour l'AR désirée. Si les sous-modules de la liste sont plus nombreux ou différents en comparaison de ceux contenus dans la liste des sous-modules prévus de l'AR-Connect, l'appareil voit une erreur de protocole et abandonner l'AR.

Après qu'un appareil a reçu un signal de Prm Begin, un sous-module affecté ne doit pas produire d'alarme jusqu'à l'Application Ready correspondante; il doit stocker ces alarmes et les émettre après l'Application Ready correspondante.

Les alarmes déjà émises à l'exception de Plug and Release doivent être acquittées par l'appareil de commande E/S. Pour basculer dans un état cohérent, l'appareil E/S patiente avec son Application Ready jusqu'à ce que toutes les alarmes en cours soient acquittées.

Les opérations de Plug and Release doivent être stockées par l'appareil de commande E/S. Elles doivent être traitées et acquittées après l'Application Ready.

Si un appareil ne prend pas en charge le service Prm Begin, il doit acquitter par l'erreur de protocole et abandonner l'AR.

Le Tableau 180 montre les paramètres du service.

Tableau 180 – Prm Begin

Nom de paramètre	req	ind	rsp	cnf
Argument	M	M(=)		
AREP	M	M(=)		
Session Key	M	M(=)		
List of Submodules	U	U(=)		
API	M	M(=)		
Slot Number	M	M(=)		
Subslot Number	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Session Key			M	M(=)
Module Diff Block			C	C(=)
List of APIs			M	M(=)
API			M	M(=)
List of Modules			M	M(=)
Slot Number			M	M(=)
Module Ident Number			M	M(=)
Module State			M	M(=)
List of Submodules			M	M(=)
Subslot Number			M	M(=)
Submodule Ident Number			M	M(=)
Submodule State			M	M(=)
Result (-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error code 1			M	M(=)
Error code 2			M	M(=)
Add Data 1			M	M(=)
Add Data 2			M	M(=)

Argument

L'argument doit acheminer les paramètres spécifiques au service relatifs à la demande de service.

AREP

Ce paramètre est l'identificateur local de l'AR souhaitée.

Session Key

La valeur du paramètre Session Key doit être augmentée de 1 pour chaque connexion par le CM Initiator (Initiateur de gestion de contexte). Le CM Responder doit utiliser la valeur pour chaque réponse ultérieure au sein de cette session.

NOTE 1 L'attribut Session Key permet au CM Initiator de détecter des erreurs de séquence au cours de la phase d'établissement et de libération d'une AR.

List of Submodules

Ce paramètre facultatif contient la liste des sous-modules qui doivent être reparamétrés.

API

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Slot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Subslot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Result (+)

Ce paramètre indique que la demande de service a réussi.

Module Diff Block

Le Module Diff Block contient les sous-modules qui ne correspondent pas la configuration prévue (erronés, manquants ou remplacés) et les sous-modules qui ne font pas partie de la relation entre applications (superordonnés verrouillés, verrouillés par un autre appareil de commande E/S ou verrouillés par le superviseur). En option, les sous-modules ayant un diagnostic et/ou de maintenance sont également indiqués dans la liste.

Le paramètre est constitué des sous-paramètres suivants:

List of APIs

Ce paramètre contient les API qui diffèrent de la configuration demandée.

API

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

List of Modules

Ce paramètre contient les modules qui diffèrent de la configuration demandée.

Slot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Module Ident Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Module State

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

List of Submodules

Cette liste de paramètres est constituée des paramètres suivants:

Subslot Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Submodule Ident Number

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Submodule State

Ce paramètre contient la valeur pour l'attribut équivalent de l'ASE.

Result(-)

Ce paramètre indique que la demande de service a échoué.