IEC 62056-6-2

Edition 4.0 2023-09

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Electricity metering data exchange – The DLMS®/COSEM suite – Part 6-2: COSEM interface classes

Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 6-2: Classes d'interfaces COSEM



IEC 62056-6-2:2023-09(en-fr)

THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2023 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 Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
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 corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
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 It also gives information on projects, replaced and withdrawn publications.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 once a month by email.

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: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

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.

Recherche de publications IEC - webstore.iec.ch/advsearchform
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 une fois par mois par email.

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

IEC Products & Services Portal - products.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 300 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 19 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC 62056-6-2

Edition 4.0 2023-09

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Electricity metering data exchange – The DLMS®/COSEM suite – Part 6-2: COSEM interface classes

Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 6-2: Classes d'interfaces COSEM

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 17.220, 35.110, 91.140.50

ISBN 978-2-8322-7344-9

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.

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

CONTENTS

FOREWORD 12
INTRODUCTION 14
1 Scope 16
2 Normative references 16
3 Terms, definitions and abbreviated terms 18
3.1 Terms and definitions related to the Image transfer process (see 4.4.6) 19
3.2 Terms and definitions related to the S-FSK PLC setup classes (see 4.10) 20
3.3 Terms and definitions related to the PRIME NB OFDM PLC setup ICs (see 4.12) 21
3.4 Terms and definitions related to the ISO/IEC 14908 setup ICs (see 4.18) 22
3.5 Terms and definitions related to ZigBee® (see 4.15) 23
3.6 Terms and definitions related to Payment metering interface classes (see 4.6) 24
3.7 Terms and definitions related to the Arbitrator IC (see 4.5.12) 28
3.8 Abbreviated terms 29
4 The COSEM interface classes 33
4.1 Basic principles 33
4.1.1 General 33
4.1.2 Referencing methods 34
4.1.3 Reserved base_names for special COSEM objects 35
4.1.4 Class description notation 35
4.1.5 Common data types 38
4.1.6 Data formats 39
4.1.7 The COSEM server model 44
4.1.8 The COSEM logical device 45
4.1.9 Information security 46
4.2 Overview of the COSEM interface classes 47
4.3 Interface classes for parameters and measurement data 53
4.3.1 Data (class_id = 1, version = 0) 53
4.3.2 Register (class_id = 3, version = 0) 54
4.3.3 Extended register (class_id = 4, version = 0) 58
4.3.4 Demand register (class_id = 5, version = 0) 60
4.3.5 Register activation (class_id = 6, version = 0) 64
4.3.6 Profile generic (class_id = 7, version = 1) 66
4.3.7 Utility tables (class_id = 26, version = 0) 72
4.3.8 Register table (class_id = 61, version = 0) 73
4.3.9 Status mapping (class_id = 63, version = 0) 76
4.3.10 Compact data (class_id: 62, version: 1) 77
4.4 Interface classes for access control and management 85
4.4.1 Overview 85
4.4.2 Client user identification 85
4.4.3 Association SN (class_id = 12, version = 4) 86
4.4.4 Association LN (class_id = 15, version = 3) 91
4.4.5 SAP assignment (class_id = 17, version = 0) 98
4.4.6 Image transfer (class_id = 18, version = 0) 98
4.4.7 Security setup (class_id = 64, version = 1) 106
4.4.8 Push interface classes and objects 112

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9 COSEM data protection (class_id = 30, version = 0) 125
4.4.10 Function control (class_id: 122, version: 0) 140
4.4.11 Array manager (class_id = 123, version = 0) 142
4.4.12 Communication port protection (class_id = 124, version = 0) 147
4.5 Interface classes for time- and event bound control 150
4.5.1 Clock (class_id = 8, version = 0) 150
4.5.2 Script table (class_id = 9, version = 0) 154
4.5.3 Schedule (class_id = 10, version = 0) 155
4.5.4 Special days table (class_id = 11, version = 0) 158
4.5.5 Activity calendar (class_id = 20, version = 0) 160
4.5.6 Register monitor (class_id = 21, version = 0) 164
4.5.7 Single action schedule (class_id = 22, version = 0) 165
4.5.8 Disconnect control (class_id = 70, version = 0) 166
4.5.9 Limiter (class_id = 71, version = 0) 170
4.5.10 Parameter monitor (class_id = 65, version = 1) 172
4.5.11 Sensor manager (class_id = 67, version = 0) 176
4.5.12 Arbitrator (class_id = 68, version = 0) 180
4.5.13 Modelling examples: tariffication and billing 184
4.6 Payment metering related interface classes 186
4.6.1 Overview of the COSEM accounting model 186
4.6.2 Account (class_id = 111, version = 0) 188
4.6.3 Credit interface class (class_id = 112, version = 0) 197
4.6.4 Charge (class_id = 113, version = 0) 208
4.6.5 Token gateway (class_id = 115, version = 0) 214
4.7 Interface classes for setting up data exchange via local ports and modems 216
4.7.1 IEC local port setup (class_id = 19, version = 1) 216
4.7.2 IEC HDLC setup (class_id = 23, version = 1) 218
4.7.3 IEC twisted pair (1) setup (class_id = 24, version = 1) 220
4.7.4 Modem configuration (class_id = 27, version = 1) 223
4.7.5 Auto answer (class_id = 28, version = 2) 224
4.7.6 Auto connect (class_id = 29, version = 2) 228
4.7.7 GPRS modem setup (class_id = 45, version = 0) 230
4.7.8 GSM diagnostic (class_id: 47, version: 2) 231
4.7.9 LTE monitoring (class_id: 151, version: 1) 233
4.8 Interface classes for setting up data exchange via M-Bus 236
4.8.1 Overview 236
4.8.2 M-Bus slave port setup (class_id = 25, version = 0) 237
4.8.3 M-Bus client (class_id = 72, version = 1) 238
4.8.4 Wireless Mode Q channel (class_id = 73, version = 1) 242
4.8.5 M-Bus master port setup (class_id = 74, version = 0) 243
4.8.6 DLMS®/COSEM server M-Bus port setup (class_id = 76, version = 0) 244
4.8.7 M-Bus diagnostic (class_id = 77, version = 0) 246
4.9 Interface classes for setting up data exchange over the Internet 248
4.9.1 TCP-UDP setup (class_id = 41, version = 0) 248
4.9.2 IPv4 setup (class_id = 42, version = 0) 250
4.9.3 IPv6 setup (class_id = 48, version = 0) 253
4.9.4 MAC address setup (class_id = 43, version = 0) 257
4.9.5 PPP setup (class_id = 44, version = 0) 257
4.9.6 SMTP setup (class_id = 46, version = 0) 262

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.7 NTP setup (class_id = 100, version = 0) 263
4.10 Interface classes for setting up data exchange using S-FSK PLC 264
4.10.1 General 264
4.10.2 Overview 265
4.10.3 S-FSK Phy&MAC set-up (class_id = 50, version = 1) 267
4.10.4 S-FSK Active initiator (class_id = 51, version = 0) 272
4.10.5 S-FSK MAC synchronization timeouts (class_id = 52, version = 0) 274
4.10.6 S-FSK MAC counters (class_id = 53, version = 0) 276
4.10.7 IEC 61334-4-32 LLC setup (class_id = 55, version = 1) 279
4.10.8 S-FSK Reporting system list (class_id = 56, version = 0) 280
4.11 Interface classes for setting up the LLC layer for ISO/IEC 8802-2 281
4.11.1 General 281
4.11.2 ISO/IEC 8802-2 LLC Type 1 setup (class_id = 57, version = 0) 281
4.11.3 ISO/IEC 8802-2 LLC Type 2 setup (class_id = 58, version = 0) 282
4.11.4 ISO/IEC 8802-2 LLC Type 3 setup (class_id = 59, version = 0) 283
4.12 Interface classes for setting up and managing DLMS®/COSEM narrowband OFDM PLC profile for PRIME networks 285
4.12.1 Overview 285
4.12.2 Mapping of PRIME NB OFDM PLC PIB attributes to COSEM IC attributes 286
4.12.3 61334-4-32 LLC SSCS setup (class_id = 80, version = 0) 288
4.12.4 PRIME NB OFDM PLC Physical layer parameters 289
4.12.5 PRIME NB OFDM PLC Physical layer counters (class_id = 81, version = 0) 289
4.12.6 PRIME NB OFDM PLC MAC setup (class_id = 82, version = 0) 290
4.12.7 NB OFDM PLC MAC functional parameters (class_id = 83 version = 0) 292
4.12.8 PRIME NB OFDM PLC MAC counters (class_id = 84, version = 0) 294
4.12.9 PRIME NB OFDM PLC MAC network administration data (class_id = 85, version = 0) 295
4.12.10 PRIME NB OFDM PLC MAC address setup (class_id = 43, version = 0) 298
4.12.11 PRIME NB OFDM PLC Application identification (class_id = 86, version = 0) 298
4.13 Interface classes for setting up and managing the DLMS®/COSEM narrowband OFDM PLC profile for G3-PLC networks 299
4.13.1 Overview 299
4.13.2 Mapping of G3-PLC PIB attributes to COSEM IC attributes 300
4.13.3 G3-PLC MAC layer counters (class_id = 90, version = 1) 301
4.13.4 G3-PLC MAC setup (class_id = 91, version = 2) 303
4.13.5 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 2) 311
4.14 Interface classes for setting up and managing DLMS®/COSEM HS-PLC ISO/IEC 12139-1 neighbourhood networks 318
4.14.1 Overview 318
4.14.2 HS-PLC ISO/IEC 12139-1 MAC setup (class_id = 140, version = 0) 318
4.14.3 HS-PLC ISO/IEC 12139-1 CPAS setup (class_id = 141, version = 0) 320
4.14.4 HS-PLC ISO/IEC 12139-1 IP SSAS setup (class_id = 142, version = 0) 320
4.14.5 HS-PLC ISO/IEC 12139-1 HDLC SSAS setup (class_id = 143, version = 0) 321
4.15 ZigBee® setup classes 322
4.15.1 Overview 322
4.15.2 ZigBee® SAS startup (class_id = 101, version = 0) 324
4.15.3 ZigBee® SAS join (class_id = 102, version = 0) 326

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.4 ZigBee® SAS APS fragmentation (class_id = 103, version = 0) 327
4.15.5 ZigBee® network control (class_id = 104, version = 0) 328
4.15.6 ZigBee® tunnel setup (class_id = 105, version = 0) 334
4.16 Interface classes for setting up and managing the DLMS®/COSEM profile for LPWAN networks 335
4.16.1 General 335
4.16.2 Generic interface classes 335
4.16.3 LPWAN specific interface classes 341
4.17 Interface classes for setting up and managing the DLMS®/COSEM profile for Wi-SUN networks 348
4.17.1 Wi-SUN setup (class_id = 95, version 0) 348
4.17.2 Wi-SUN diagnostic (class_id = 96, version 0) 352
4.17.3 RPL diagnostic (class_id = 97, version 0) 356
4.17.4 MPL diagnostic (class_id = 98, version 0) 358
4.18 Interface classes for setting up and managing the DLMS®/COSEM profile for ISO/IEC 14908 PLC networks 360
4.18.1 General 360
4.18.2 ISO/IEC 14908 identification (class_id = 130, version = 0) 360
4.18.3 ISO/IEC 14908 protocol setup (class_id = 131, version = 0) 361
4.18.4 ISO/IEC 14908 protocol status (class_id = 132, version = 0) 361
4.18.5 ISO/IEC 14908 diagnostic (class_id = 133, version = 0) 363
5 Previous versions of interface classes 366
5.1 General 366
5.1.1 New versions of interface classes 366
5.1.2 New interface classes 366
5.1.3 Removal of interface classes 366
5.2 Previous versions of interface classes – general 366
5.3 Previous versions of interface classes for parameters and measurement data 366
5.3.1 Profile generic (class_id = 7, version = 0) 366
5.3.2 Compact data (class_id = 62, version = 0) 370
5.4 Previous versions of interface classes for access control and management 373
5.4.1 Association SN (class_id = 12, version = 0) 373
5.4.2 Association SN (class_id = 12, version = 1) 375
5.4.3 Association SN (class_id = 12, version = 2) 378
5.4.4 Association SN (Class_id = 12, version =3) 382
5.4.5 Association LN (class_id = 15, version = 0) 387
5.4.6 Association LN (class_id = 15, version = 1) 392
5.4.7 Association LN (class_id = 15, version = 2) 397
5.4.8 Security setup (class_id = 64, version = 0) 403
5.4.9 Push Setup (class_id = 40, version = 0) 405
5.4.10 Push Setup (class_id = 40, version = 1) 410
5.5 Previous versions of interface classes for time- and event-bound control 417
5.5.1 Parameter monitor (class_id = 65, version = 0) 417
5.6 Previous versions of payment metering related interface classes 418
5.7 Previous versions of interface classes for setting up data exchange via local ports and modems 419
5.7.1 IEC local port setup (class_id = 19, version = 0) 419
5.7.2 IEC HDLC setup, (class_id = 23, version = 0) 420
5.7.3 IEC twisted pair (1) setup (class_id = 24, version = 0) 422

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.4 PSTN modem configuration (class_id = 27, version = 0) 424
5.7.5 Auto answer (class_id = 28, version = 0) 426
5.7.6 PSTN auto dial (class_id = 29, version = 0) 428
5.7.7 Auto connect (class_id = 29, version = 1) 429
5.7.8 GSM diagnostic (class_id = 47, version = 0) 431
5.7.9 GSM diagnostic (class_id: 47, version: 1) 433
5.7.10 LTE monitoring (class_id: 151, version: 0) 435
5.8 Previous versions of interface classes for setting up data exchange via M-Bus 436
5.8.1 M-Bus client (class_id = 72, version = 0) 436
5.9 Previous versions of interface classes for setting up data exchange over the internet 441
5.10 Previous versions of interface classes for data exchange using S-FSK PLC 441
5.10.1 S-FSK Phy&MAC setup (class_id = 50, version = 0) 441
5.10.2 S-FSK IEC 61334-4-32 LLC setup (class_id = 55, version = 0) 446
5.11 Previous versions of interface classes for setting up the LLC layer for ISO/IEC 8802-2 447
5.12 Previous versions of interface classes for setting up and managing DLMS®/COSEM narrowband OFDM PLC profile for PRIME networks 447
5.13 Previous versions of interface classes for setting up and managing DLMS®/COSEM narrowband OFDM PLC profile for G3-PLC networks 447
5.13.1 Mapping of G3-PLC IB attributes to COSEM IC attributes (Original version) 447
5.13.2 G3 NB OFDM PLC MAC layer counters (class_id = 90, version = 0) 449
5.13.3 G3 NB OFDM PLC MAC setup (class_id = 91, version = 0) 450
5.13.4 G3 NB OFDM PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 0) 454
5.13.5 Mapping of G3-PLC PIB attributes to COSEM IC attributes (previous version) 460
5.13.6 G3-PLC MAC setup (class_id = 91, version = 1) 462
5.13.7 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 1) 468
5.14 Previous versions of interface classes for setting up and managing DLMS®/COSEM HS-PLC ISO/IEC 12139-1 neighbourhood networks 474
5.15 Previous versions of ZigBee® setup classes 474
6 Relation to OBIS 474
6.1 General 474
6.2 Abstract COSEM objects 475
6.2.1 Use of value group C 475
6.2.2 Data of historical billing periods 476
6.2.3 Billing period values / reset counter entries 478
6.2.4 Other abstract general purpose OBIS codes 478
6.2.5 Clock objects (class_id = 8) 479
6.2.6 Modem configuration and related objects 479
6.2.7 Script table objects (class_id = 9) 480
6.2.8 Special days table objects (class_id = 11) 481
6.2.9 Schedule objects (class_id = 10) 481
6.2.10 Activity calendar objects (class_id = 20) 482
6.2.11 Register activation objects (class_id = 6) 482
6.2.12 Single action schedule objects (class_id = 22) 482
6.2.13 Register monitor objects (class_id = 21) 483
6.2.14 Parameter monitor objects (class_id = 65) 483

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.15 Limiter objects (class_id = 71) 483
6.2.16 Array manager objects (class_id = 123) 483
6.2.17 Payment metering related objects 483
6.2.18 IEC local port setup objects (class_id = 19) 484
6.2.19 Standard readout profile objects (class_id = 7) 484
6.2.20 IEC HDLC setup objects (class_id = 23) 485
6.2.21 IEC twisted pair (1) setup objects (class_id = 24) 485
6.2.22 Objects related to data exchange over M-Bus 486
6.2.23 Objects to set up data exchange over the Internet 487
6.2.24 Objects to set up Push Setup (class_id = 40) 488
6.2.25 Objects for setting up data exchange using S-FSK PLC 489
6.2.26 Objects for setting up the ISO/IEC 8802-2 LLC layer 489
6.2.27 Objects for data exchange using narrowband OFDM PLC for PRIME networks 490
6.2.28 Objects for data exchange using narrow-band OFDM PLC for G3-PLC networks 490
6.2.29 ZigBee® setup objects 491
6.2.30 Objects for setting up and managing data exchange using ISO/IEC 14908 PLC networks 491
6.2.31 Objects for data exchange using HS-PLC ISO/IEC 12139-1 ISO/EC 12139-1 networks 491
6.2.32 Objects for data exchange using Wi-SUN networks 492
6.2.33 Association objects (class_id = 12, 15) 492
6.2.34 SAP assignment object (class_id = 17) 493
6.2.35 COSEM logical device name object 493
6.2.36 Information security related objects 493
6.2.37 Image transfer objects (class_id = 18) 494
6.2.38 Function control objects (class_id = 122) 494
6.2.39 Communication port protection objects (class_id = 124) 494
6.2.40 Utility table objects (class_id = 26) 494
6.2.41 Compact data objects (class_id = 62) 495
6.2.42 Device ID objects 495
6.2.43 Metering point ID objects 496
6.2.44 Parameter changes and calibration objects 496
6.2.45 I/O control signal objects 496
6.2.46 Disconnect control objects (class_id = 70) 496
6.2.47 Arbitrator objects (class_id = 68) 497
6.2.48 Status of internal control signals objects 497
6.2.49 Internal operating status objects 497
6.2.50 Battery entries objects 498
6.2.51 Power failure monitoring objects 498
6.2.52 Operating time objects 499
6.2.53 Environment related parameters objects 499
6.2.54 Status register objects 499
6.2.55 Event code objects 499
6.2.56 Communication port log parameter objects 500
6.2.57 Consumer message objects 500
6.2.58 Currently active tariff objects 500
6.2.59 Event counter objects 500
6.2.60 Profile entry digital signature objects 501

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.61 Profile entry counter objects 501
6.2.62 Meter tamper event related objects 501
6.2.63 Error register objects 502
6.2.64 Alarm register, Alarm filter and Alarm descriptor objects 502
6.2.65 General list objects 503
6.2.66 Event log objects (class_id 7) 503
6.2.67 Inactive objects 504
6.3 Electricity related COSEM objects 504
6.3.1 Value group D definitions 504
6.3.2 ID numbers – Electricity 504
6.3.3 Billing period values / reset counter entries 505
6.3.4 Other electricity related general purpose objects 505
6.3.5 Measurement algorithm 506
6.3.6 Metering point ID (electricity related) 508
6.3.7 Electricity related status objects 508
6.3.8 List objects – Electricity (class_id = 7) 508
6.3.9 Threshold values 509
6.3.10 Register monitor objects (class_id = 21) 510
6.4 Coding of OBIS identifications 510
Annex A (informative) Additional information on Auto answer and Auto connect ICs 512
Annex B (informative) Additional information to M-Bus client (class_id = 72, version 1) 514
Annex C (informative) Additional information on IPv6 setup class (class_id = 48, version = 0) 516
C.1 General 516
C.2 IPv6 addressing 516
C.3 IPv6 header format 517
C.4 IPv6 header extensions 519
C.4.1 Overview 519
C.4.2 Hop-by-Hop options 519
C.4.3 Destination options 520
C.4.4 Routing options 520
C.4.5 Fragment options 520
C.4.6 Security options 520
Annex D (informative) Overview of the narrow-band OFDM PLC technology for PRIME networks 521
Annex E (informative) Overview of the narrow-band OFDM PLC technology for G3-PLC networks 522
Annex F (informative) Significant technical changes with respect to IEC 62056-6-2, Edition 3.0:2017 523
Bibliography 524

Figure 1 – The meaning of the definitions concerning the Image 19
Figure 2 – An interface class and its instances 34
Figure 3 – The COSEM server model 44
Figure 4 – Combined metering device 45
Figure 5 – Overview of the interface classes – Part 1 48
Figure 6 – Overview of the interface classes – Part 2 49
Figure 7 – Overview of the interface classes – Part 3 50

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 8 – The time attributes when measuring sliding demand 60
Figure 9 – The attributes in the case of block demand 61
Figure 10 – The attributes in the case of sliding demand (number of periods = 3) 61
Figure 11 – Image transfer process flow chart 104
Figure 12 – COSEM model of push operation 113
Figure 13 – Push windows and delays 115
Figure 14 – COSEM model of data protection 126
Figure 15 – Example: Read *protection_buffer* attribute 128
Figure 16 – Example of managing an array 143
Figure 17 – The generalized time concept 151
Figure 18 – State diagram of the Disconnect control IC 167
Figure 19 – Definition of upper and lower thresholds 180
Figure 20 – COSEM tariffication model (example) 185
Figure 21 – COSEM billing model (example) 185
Figure 22 – Outline Account model 187
Figure 23 – Diagram of attribute relationships 188
Figure 24 – Credit States when priority >0 198
Figure 25 – Operation of current_credit_status flags 201
Figure 26 – Interaction of current_credit_amount and available_credit with Token "Credit" and Emergency "Credit" 207
Figure 27 – Object model of DLMS®/COSEM servers 265
Figure 28 – Object model of DLMS®/COSEM servers 286
Figure 29 – Example of a ZigBee® network 323
Figure 30 – Push windows and delays 406
Figure 31 – Data of historical billing periods – example with module 12, VZ = 5 477
Figure A.1 – Network connectivity example for a GSM/GPRS network 512
Figure B.1 – Encryption key status diagram 514
Figure C.1 – IPv6 address formats 517
Figure C.2 – IPv6 header format 518
Figure C.3 – Traffic class parameter format 518

Table 1 – Reserved base_names for SN referencing 35
Table 2 – Interface class overview 35
Table 3 – Common data types 38
Table 4 – List of interface classes by class_id 50
Table 5 – Enumerated values for physical units 56
Table 6 – Examples for scaler_unit 58
Table 7 – Parameters for selective access to the buffer attribute 70
Table 8 – Parameters for selective access to the buffer attribute 72
Table 9 – Daily billing data 81
Table 10 – Attributes of the "Compact data" object 81
Table 11 – A-XDR encoding of the data (SEQUENCE OF Get-Data-Result) 82
Table 12 – Diagnostic and Alarm data 82

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 13 – Attributes of the “Compact data” object 83
Table 14 – Encoding the data read from the buffer attribute of a “Profile generic” object 83
Table 15 – Logbook data 83
Table 16 – Attributes of the “Compact data” object 84
Table 17 – Attributes of the “Compact data” object 84
Table 18 – A-XDR encoding of the data read from the *buffer* attribute 85
Table 19 – Parameters for selective access to the object_list and access_rights_list attribute 89
Table 20 – Parameters for selective access to the object_list attribute 93
Table 21 – Encoding of selective access parameters with data_index 124
Table 22 – Key information required to establish data protection keys 135
Table 23 – Protection parameters of *protection_parameters_get* attribute 136
Table 24 – Protection parameters of *protection_parameters_set* attribute 137
Table 25 – Protection parameters of *get_protected_attributes* method 138
Table 26 – Protection parameters of *set_protected_attributes* method 139
Table 27 – Protection parameters of *invoke_protected_method* method 140
Table 28 – Example values for NCA and CLT 149
Table 29 – Schedule 155
Table 30 – Special days table 155
Table 31 – Disconnect control IC – states and state transitions 168
Table 32 – Explicit presentation of threshold value arrays 180
Table 33 – Explicit presentation of action_sets 180
Table 34 – Credit states 198
Table 35 – Credit state transitions 199
Table 36 – ADS address elements 222
Table 37 – Fatal error register 223
Table 38 – Mapping IEC 61334-4-512:2001 MIB variables to COSEM IC attributes / methods 266
Table 39 – MAC addresses in the S-FSK profile 272
Table 40 – Mapping of PRIME NB OFDM PLC PIB attributes to COSEM IC attributes 287
Table 41 – Mapping of G3-PLC IB attributes to COSEM IC attributes 300
Table 42 – Use of ZigBee® setup COSEM interface classes 323
Table 43 – C/D Rule 1 339
Table 44 – Parameters for selective access to the object_list attribute 376
Table 45 – Parameters for selective access to the *object_list* and *access_rights_list* attribute 381
Table 46 – Parameters for selective access to the object_list attribute 385
Table 47 – Parameters for selective access to the *object_list* attribute 391
Table 48 – Encoding of selective access parameters with data_index 410
Table 49 – ADS address elements 423
Table 50 – Mapping of G3-PLC IB attributes specified in ITU-T G.9903:2017 to COSEM IC attributes 448
Table 51 – Mapping of G3-PLC IB attributes to COSEM IC attributes 461
Table 52 – Use of value group C for abstract objects in the COSEM context 475

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 53 – Representation of various values by appropriate ICs 504
Table 54 – Measuring algorithms – enumerated values 507
Table 55 – Threshold objects, electricity 509
Table 56 – Register monitor objects, electricity 510
Table B.1 – Encryption key is preset in the slave and cannot be changed 515
Table B.2 – Encryption key is preset in the slave and new key is set after installation 515
Table B.3 – Encryption key is not preset in the slave, but can be set, case a) 515
Table B.4 – Encryption key is not preset in the slave, but can be set, case b) 515
Table C.1 – IPv6 header vs. IPv6 IC 519
Table C.2 – Optional IPv6 header extensions vs. IPv6 IC 519

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

ELECTRICITY METERING DATA EXCHANGE – THE DLMS®/COSEM SUITE –

Part 6-2: COSEM interface classes

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.

IEC 62056-6-2 has been prepared by IEC technical committee 13: Electrical energy measurement and control. It is an International Standard.

This fourth edition cancels and replaces the third edition of IEC 62056-6-2 published in 2017. It constitutes a technical revision.

The significant technical changes with respect to the previous edition are listed in Annex F (Informative).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The text of this International Standard is based on the following documents:

Draft	Report on voting
13/1891/FDIS	13/1906/RVD

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

A list of all the parts in the IEC 62056 series, published under the general title *Electricity metering data exchange – The DLMS®/COSEM suite*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

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

IMPORTANT – The "colour inside" logo on the cover page of this document 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.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

INTRODUCTION

This fourth edition of IEC 62056-6-2 has been prepared by IEC TC13 WG14 with a significant contribution of the DLMS® User Association, its A-type liaison partner.

This edition is in line with the DLMS® UA Blue Book Edition 14. The main new features are the “Array manager” IC, version 1 of the “Compact data” IC, version 1 of the “GSM diagnostic” IC, the “LTE monitoring” IC, the “NTP setup” IC, the HS-PLC setup ICs and the related new OBIS codes.

Object modelling and data identification

Driven by the business needs of the energy market participants – generally in a liberalized, competitive environment – and by the desire to manage natural resources efficiently and to involve the consumers, the utility meter became part of an integrated metering, control and billing system. The meter is not any more a simple data recording device but it relies critically on communication capabilities. Ease of system integration, interoperability and data security are important requirements.

COSEM, the *Companion Specification for Energy Metering*, addresses these challenges by looking at the utility meter as part of a complex measurement and control system. The meter has to be able to convey measurement results from the metering points to the business processes which use them. It also has to be able to provide information to the consumer and manage consumption and eventually local generation.

COSEM achieves this by using *object modelling* techniques to model all functions of the meter, without making any assumptions about which functions need to be supported, how those functions are implemented and how the data are transported. The formal specification of COSEM interface classes forms a major part of COSEM.

To process and manage the information it is necessary to uniquely identify all data items in a manufacturer-independent way. The definition of OBIS, the *Object Identification System* is another essential part of COSEM. It is based on DIN 43863-3:1997, *Electricity meters – Part 3: Tariff metering device as additional equipment for electricity meters – EDIS – Energy Data Identification System*. The set of OBIS codes has been considerably extended over the years to meet new needs.

COSEM models the utility meter as a *server* application – see 4.1.7 – used by *client* applications that retrieve data from, provide control information to, and instigate known actions within the meter via controlled access to the COSEM objects. The *clients* act as agents for third parties, i.e. the business processes of energy market participants.

The standardized COSEM interface classes form an extensible library. Manufacturers use elements of this library to design their products that meet a wide variety of requirements.

The server offers means to retrieve the functions supported, i.e. the COSEM objects instantiated. The objects can be organized to *logical devices and application associations* and to provide specific access rights to various clients.

The concept of the standardized interface class library provides different users and manufacturers with a maximum of diversity while ensuring interoperability.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this International Standard may involve the use of a maintenance service concerning the stack of protocols on which the present standard IEC 62056-6-2 is based.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The IEC takes no position concerning the evidence, validity and scope of this maintenance service.

The provider of the maintenance service has assured the IEC that he is willing to provide services under reasonable and non-discriminatory terms and conditions for applicants throughout the world. In this respect, the statement of the provider of the maintenance service is registered with the IEC. Information may be obtained from:

DLMS[1] User Association
www.dlms.com

Acknowledgement

The actual document has been established by the WG Maintenance of the DLMS® UA.

Subclauses 4.4.7 and 4.4.9 are based on parts of NIST documents. Reprinted courtesy of the National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce. Not copyrightable in the United States.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

[1] Device Language Message Specification.

ELECTRICITY METERING DATA EXCHANGE – THE DLMS®/COSEM SUITE –

Part 6-2: COSEM interface classes

1 Scope

This part of IEC 62056 specifies a model of a meter as it is seen through its communication interface(s). Generic building blocks are defined using object-oriented methods, in the form of interface classes to model meters from simple up to very complex functionality.

Annexes A to F (informative) provide additional information related to some interface classes.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 13757-2:2004, *Communication system for and remote reading of meters – Part 2: Physical and link layer*

EN 13757-3:2004, *Communication systems for and remote reading of meters – Part 3: Dedicated application layer*

NOTE This standard is referenced in the “M-Bus client setup” interface class version 0.

EN 13757-3:2013, *Communication systems for and remote reading of meters – Part 3: Dedicated application layer*

NOTE This standard is referenced in the M-Bus client setup interface class version 1.

EN 13757-4:2013, *Communication system for and remote reading of meters – Part 4: Wireless meter (Radio meter reading for operation in SRD bands)*

EN 13757-5:2015, *Communication systems for meters – Part 5: Wireless M-Bus relaying*

IEC 61334-4-32:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 32: Data link layer – Logical link control (LLC)*

IEC 61334-4-41:1996, *Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 41: Application protocols – Distribution line message specification*

IEC 61334-4-511:2000, *Distribution automation using distribution line carrier systems – Part 4-511: Data communication protocols – Systems management – CIASE protocol*

IEC 61334-4-512:2001, *Distribution automation using distribution line carrier systems – Part 4-512: Data communication protocols – System management using profile 61334-5-1 – Management Information Base (MIB)*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC 61334-5-1:2001, *Distribution automation using distribution line carrier systems – Part 5-1: Lower layer profiles – The spread frequency shift keying (S-FSK) profile*

IEC TR 62055-21:2005, *Electricity metering – Payment systems – Part 21: Framework for standardization*

IEC 62056-21:2002, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange*

IEC 62056-31:1999, *Electricity metering – Data exchange for meter reading, tariff and load control – Part 31: Using local area networks on twisted pair with carrier signalling*

NOTE This Edition is referenced in the interface class "IEC twisted pair (1) setup" (class_id: 24, version: 0).

IEC 62056-3-1:2013, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 3-1: Use of local area networks on twisted pair with carrier signalling*

NOTE This Edition is referenced in the interface class "IEC twisted pair (1) setup" (class_id: 24, version: 1).

IEC 62056-3-1:2021, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 3-1: Use of local area networks on twisted pair with carrier signalling*

IEC 62056-46:2002/AMD1:2006 ,*Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data link layer using HDLC protocol*

IEC 62056-5-3:2023, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 5-3: DLMS®/COSEM application layer*

IEC 62056-6-1:2023, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 6-1: Object identification system (OBIS)*

IEC 62056-7-3:2017, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 7-3: Wired and wireless M-Bus communication profiles for local and neighbourhood networks*

IEC 62056-8-3:2013, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 8-3: Communication profile for PLC S-FSK neighbourhood networks*

IEC 62056-8-4:2018, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 8-4: Communication profiles for narrow-band OFDM PLC PRIME neighbourhood networks*

IEC 62056-8-6:2017, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 8-6: High speed PLC ISO/IEC 12139-1 profile for neighbourhood networks*

IEC 62056-8-8:2020, *Electricity metering data exchange – The DLMS®/COSEM suite – Part 8-8: Communication profile for ISO/IEC 14908 series networks*

IEEE 802.15.4:2006, *Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*

NOTE This standard is also available as ISO/IEC/IEEE 8802-15-4:2010.

IETF STD 51, *The Point-to-Point Protocol (PPP)*, 1994. (Also RFC 1661, RFC 1662)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

ISO/IEC 8802-2:1998, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 2: Logical Link Control*

ISO/IEC 12139-1:2009, *Information technology – Telecommunications and information exchange between systems – Powerline communication (PLC) – High speed PLC medium access control (MAC) and physical layer (PHY) – Part 1: General requirements*

ISO/IEC 14908-1:2012, *Information technology – Control network protocol – Part 1: Protocol stack*

ISO/IEC 60559:2020, *Information technology – Microprocessor Systems – Floating-Point arithmetic*

ISO 4217, *Codes for the representation of currencies*

ITU-T E.212 (05.2008), *Series E: Overall network operation,telephone service, service operation and human factors – International operation – Maritime mobile service and public land mobile service – The international identification plan for public networks and subscriptions*

ITU-T G.9903 Amd. 1:2013, *Series G: Transmission systems and media, digital systems and networks – Access networks – In premises networks – Narrow-band orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks*

NOTE This Recommendation is referenced in version 0 of the G3-PLC setup classes.

ITU-T G.9903:2014, *Series G: Transmission systems and media, digital systems and networks – Access networks – In premises networks – Narrow-band orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks*

NOTE This Recommendation is referenced in version 1 of the G3-PLC setup classes.

ITU-T G.9903 Amd. 1:2017, *Series G: Transmission systems and media, digital systems and networks – Access networks – In premises networks – Narrow-band orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks*

ITU-T G.9904:2012, *Series G: Transmission systems and media, digital systems and networks – Access networks – In premises networks – Narrow-band orthogonal frequency division multiplexing power line communication transceivers for PRIME networks*

ETSI GSM 05.08:1996, *Digital cellular telecommunications system (Phase 2+); Radio subsystem link control*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3 Terms, definitions and abbreviated terms

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at http://www.electropedia.org/
- ISO Online browsing platform: available at http://www.iso.org/obp

3.1 Terms and definitions related to the Image transfer process (see 4.4.6)

3.1.1
Image
binary data of a specified size

Note 1 to entry: An Image can be seen as a container. It may consist of one or multiple elements (image_to_activate) which are transferred, verified and activated together.

3.1.2
ImageSize
size of the whole Image to be transferred

Note 1 to entry: ImageSize is expressed in octets.

3.1.3
ImageBlock
part of the Image of size ImageBlockSize

Note 1 to entry: The Image is transferred in ImageBlocks. Each block is identified by its ImageBlockNumber.

3.1.4
ImageBlockSize
size of ImageBlock expressed in octets

3.1.5
ImageBlockNumber
identifier of an ImageBlock. ImageBlocks are numbered sequentially, starting from 0

The meaning of the definitions above is illustrated in Figure 1.

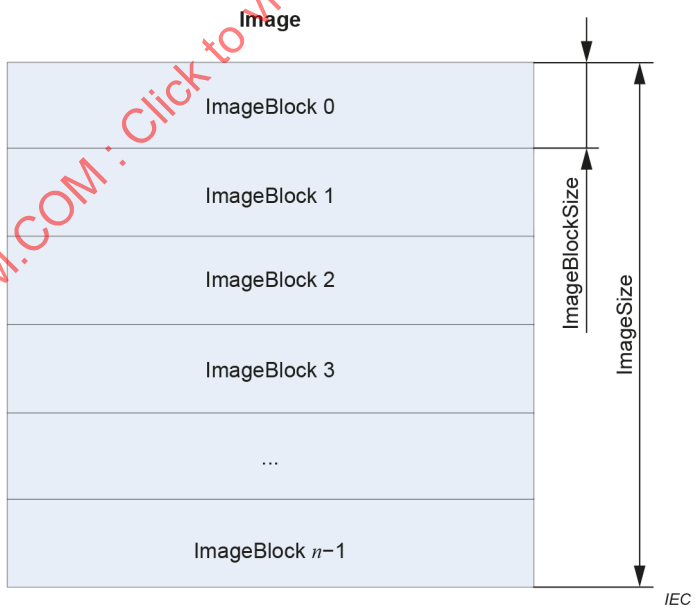


Figure 1 – The meaning of the definitions concerning the Image

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.2 Terms and definitions related to the S-FSK PLC setup classes (see 4.10)

3.2.1
initiator
user-element of a client System Management Application Entity (SMAE)

Note 1 to entry: The initiator uses the CIASE and xDLMS ASE and is identified by its system title.

[SOURCE: IEC 61334-4-511:2000, 3.8.1]

3.2.2
active initiator
initiator which issues or has last issued a CIASE Register request when the server is in the unconfigured state

[SOURCE: IEC 61334-4-511:2000, 3.9.1]

3.2.3
new system
server system which is in the unconfigured state: its MAC address equals "NEW-address"

[SOURCE: IEC 61334-4-511:2000, 3.9.3]

3.2.4
new system title
system-title of a new system

Note 1 to entry: This is the system title of a system, which is in the new state.

[SOURCE: IEC 61334-4-511:2000, 3.9.4]

3.2.5
registered system
server system which has an individual valid MAC address (therefore, different from "NEW Address", see IEC 61334-5-1:2001: Medium Access Control)

[SOURCE: IEC 61334-4-511:2000, 3.9.5]

3.2.6
reporting system
server system which issues a DiscoverReport

[SOURCE: IEC 61334-4-511:2000, 3.9.6, modified to correct an error in IEC 61334-4-511]

3.2.7
sub-slot
time needed to transmit two bytes by the physical layer

Note 1 to entry: Timeslots are divided to sub-slots in the RepeaterCall mode of the physical layer.

3.2.8
timeslot
time needed to transmit a physical frame

Note 1 to entry: As specified in IEC 61334-5-1:2001, 3.3.1, a physical frame comprises 2 bytes preamble, 2 bytes start subframe delimiter, 38 bytes PSDU and 3 bytes pause.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.3 Terms and definitions related to the PRIME NB OFDM PLC setup ICs (see 4.12)

Definitions related to the physical layer

3.3.1
base node
the master node, which controls and manages the resources of a subnetwork

[SOURCE: ITU-T G.9904:2012, 3.2.1]

3.3.2
beacon slot
the location of the beacon PDU within a frame

[SOURCE: ITU-T G.9904:2012, 3.2.2]

3.3.3
node
any one element of a subnetwork, which is able to transmit to and receive from other subnetwork elements

[SOURCE: ITU-T G.9904:2012, 3.2.9]

3.3.4
registration
the process by which a service node is accepted as member of the subnetwork and allocated a LNID

[SOURCE: ITU-T G.9904:2012, 3.2.12]

3.3.5
service node
any one node of a subnetwork, which is not a base node

[SOURCE: ITU-T G.9904:2012, 3.2.13]

3.3.6
subnetwork
a set of elements that can communicate by complying with this specification and share a single base node

[SOURCE: ITU-T G.9904:2012, 3.2.15]

Definitions related to the MAC layer

3.3.7
disconnected state
<of a service node>this is the initial functional state for all service nodes. When disconnected, a service node is not able to communicate data or switch other nodes' data; its main function is to search for a subnetwork within its reach and try to register on it

[SOURCE: ITU-T G.9904:2012, 8.1]

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.3.8
terminal state
<of a service node>when in this functional state a service node is able to establish connections and communicate data, but it is not able to switch other nodes' data

[SOURCE: ITU-T G.9904:2012 8.1]

3.3.9
switch state
<of a service node>when in this functional state a service node is able to perform all Terminal functions. Additionally, it is able to forward data to and from other nodes in the same subnetwork. It is a branch point on the tree structure

[SOURCE: ITU-T G.9904:2012, 8.1]

3.3.10
promotion
the process by which a service node is qualified to switch (repeat, forward) data traffic from other nodes and act as a branch point on the subnetwork tree structure. A successful promotion represents the transition between Terminal and Switch state. When a service node is in the Disconnected state, it cannot directly transition to Switch state

[SOURCE: ITU-T G.9904:2012, 8.1]

3.3.11
demotion
the process by which a service node ceases to be a branch point on the subnetwork tree structure. A successful demotion represents the transition between Switch and Terminal state

[SOURCE: ITU-T G.9904:2012, 8.1]

3.4 Terms and definitions related to the ISO/IEC 14908 setup ICs (see 4.18)

3.4.1
domain
logical network that is a unit for addressing

Note 1 to entry: Subnet (see below) and node addresses are assigned by the administrator responsible for the domain, and they have meaning only in the context of that domain.

Note 2 to entry: All nodes shall be in the same domain to be able to address each other.

3.4.2
node
abstraction for a physical node that represents the highest degree of address resolvability on a network

Note 1 to entry: A node is identified (addressed) within a subnet by its (logical) node identifier. A physical node may belong to more than one subnet; when it does, it is assigned one (logical) node number for each subnet to which it belongs. A physical node may belong to at most two subnets; these subnets shall be in different domains. A node may also be identified (absolutely) within a network by its Unique_Node_ID.

3.4.3
subnet
set of nodes accessible through the same link layer protocol

3.4.4
transaction
sequence of messages that are correlated together

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.5 Terms and definitions related to ZigBee® (see 4.15)

NOTE Terms marked with * are from the ZigBee® Specification.

3.5.1
CAD
Consumer Access Device; a ZigBee® gateway device that acts like an IHD within the ZigBee® network, but has an additional connection to a different network (i.e. WiFi)

3.5.2
IHD
in Home Display; a device that has a screen for the displaying of Energy information to the consumer

3.5.3
install code
Hashed (via MMO) Pre-Configured Linked Key (PCLK) that is provided to a Trust Center via out-of-band communications. A new device wishing to join the network would need to send this install code to the Trust Center, which would allow the Trust Center to execute the joining process, using this install code as part of the security information

3.5.4
link key *
key that is shared exclusively between two, and only two, peer application-layer entities within a PAN

3.5.5
MAC address/IEEE address
these are used synonymously to represent the EUI-64 code allocated to the ZigBee® Radio

3.5.6
ZigBee®
specification for a suite of high level communication protocols used to create personal area networks built from small, low-power digital radios. ZigBee® is based on an IEEE 802.15 standard. Though low-powered, ZigBee® devices often transmit data over longer distances by passing data through intermediate devices to reach more distant ones, creating a mesh network

3.5.7
ZigBee® client
similar to the role of the DLMS®/COSEM Client

Note 1 to entry: For a greater understanding of the interaction between the client and server the ZigBee® PRO specification should be read.

3.5.8
ZigBee® coordinator *
IEEE 802.15.4-2006 PAN coordinator that is the principal controller of an IEEE 802.15.4-2006-based network that is responsible for network formation. The PAN coordinator shall be a full function device (FFD)

3.5.9
ZigBee® cluster
set of message types related to a certain device function (e.g. metering, ballast control)

3.5.10
ZigBee® mirror
device which echoes data being published by a battery operated ZigBee® device, allowing other network actors to obtain data while the battery operated device is unavailable due to power saving

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.5.11
ZigBee® PRO
alternative name for the ZigBee® 2007 protocol. ZigBee® 2007, now the current stack release, contains two stack profiles, stack profile 1 (simply called ZigBee®), for home and light commercial use, and stack profile 2 (called ZigBee® PRO). ZigBee® PRO offers more features, such as multi-casting, many-to-one routing and high security with Symmetric-Key Key Exchange (SKKE), while ZigBee® (stack profile 1) offers a smaller footprint in RAM and flash. Both offer full mesh networking and work with all ZigBee® application profiles

3.5.12
ZigBee® router *
IEEE 802.15.4-2006 FFD participating in a ZigBee® network, which is not the ZigBee® coordinator, but may act as an IEEE 802.15.4-2006 coordinator within its personal operating space, that is capable of routing messages between devices and supporting associations

3.5.13
ZigBee® server
similar to the role of the DLMS®/COSEM Server

Note 1 to entry: For a greater understanding of the interaction between the client and server the ZigBee® PRO specification should be read.

3.5.14
ZigBee® Trust Center *
device trusted by devices within a ZigBee® network to distribute keys for the purpose of network and end-to-end application configuration management

3.6 Terms and definitions related to Payment metering interface classes (see 4.6)

3.6.1
account
statement of the credits and charges of an individual with reference to a contractual relationship between the said individual and another party; in this case a utility service provider

3.6.2
available
(credit), total value that may be decremented by charges without further action

3.6.3
charge
representation of a financial liability on an account

Note 1 to entry: Within this specification charges are modelled in the form of “Charge” objects that define the amount due, collection mechanism, collection periodicity, collection amount and other relevant variables.

Note 2 to entry: There may be one or more instalments payable and their size may be determined explicitly or in terms of a rate of payment per unit of time or of consumption.

Note 3 to entry: Charges may also be levied as a fixed amount per vend.

3.6.4
credit mode
mode of operation of a meter in a payment system that does not require payment for the consumption in advance

3.6.5
collect
take payment of an instalment of a charge, accounting for the collection amount determined by the *unit_charge_active* attribute of the “Charge” object

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.6.6
commodity
utility product delivered to a consumer at a service point on their premises under a contract of supply such as electricity, gas, water, and heat

3.6.7
enabled
when used in the context of “Credit” or “Charge” types; means that the “Credit” or “Charge” type appears in the *credit_reference_list* or *charge_reference_list* respectively of the “Account” object

3.6.8
emergency credit
amount of credit administered in a payment metering system working in prepayment mode, representing a short term loan to the consumer

Note 1 to entry: This is a feature of some payment metering systems in which the consumer is able to obtain a limited amount of credit as a short-term loan, often mediated locally by the prepayment unit itself. The word “emergency” indicates urgent need rather than disaster.

3.6.9
Enterprise Resource Planning (ERP) system
Back Office System
computer system carrying out the business processing of an organisation (such as an energy supplier), as distinct from the communications system. (See also Head End System)

3.6.10
friendly credit
period of time with a configurable start and end point, where the meter will not disconnect supply regardless of the status of the *available_credit*. Also known as non-disconnect period

Note 1 to entry: This function is used in circumstances where it would be inconvenient to obtain needed credit (for example, at night or in the case of a frail elderly consumer).

3.6.11
Head End System
HES
computer system, connected by a communications network to a population of intelligent devices, whose job is the control and coordination of information flows to and from those devices, typically on behalf of a separate ERP (“back office”) system

3.6.12
Home Area Network
HAN
communications network constructed with the principal aim of connecting devices in one premises

3.6.13
in use
state of a “Credit” object that, at the point of query, has a positive *current_credit_amount* and that Credit is being consumed by some active Charges represented by “Charge” objects

Note 1 to entry: When the current_credit_amount reaches zero, the credit status becomes exhausted.

3.6.14
load limiting
mode of operation of some payment metering systems (not necessarily in prepayment mode) in which the consumer is able to draw on a supply provided they do not exceed a configured level of demand

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Note 1 to entry: The implied purpose is for management of the consumer's finances: where demand is subject to limitation for the benefit of the generation or distribution system the term "load management" is more often used.

3.6.15
local communications
mechanism of communicating with the meter over some media, within the vicinity of that meter such as over a HAN or optical port

3.6.16
manual entry
entering of a token to the payment metering installation via means of a manual process

3.6.17
managed payment mode
specialisation of credit mode that allows operation of an Account, Credit and possibly Charges in a meter where the payment for the service is received by the utility after the service has been consumed

Note 1 to entry: When in managed payment mode tokens are not normally used, however the credit is adjusted using the methods in the "Credit" object.

Note 2 to entry: The meter is allowed to go into an allowable amount of debt before being credited from the client in line with a received cash payment by the utility.

Note 3 to entry: In this example cash is used as a generic term for a real life payment of currency to the utility which could be executed as legal tender, automated electronic transfer, etc.

3.6.18
payment metering installation
set of payment metering equipment installed and ready for use at a consumer's premise

Note 1 to entry: This includes mounting the equipment as appropriate, and where a multi-device installation is involved, the connection of each unit of equipment as appropriate. It also includes the connection of utility supply network to each supply interface, the connection of the consumer's load interface, and the commissioning of the equipment into an operational state as a payment metering installation.

3.6.19
prepayment mode
mode of operation of a meter in a payment system, whereby the consumer pays for service in advance of consumption

3.6.20
post-payment
method of operation of a payment system whereby a consumer may consume service before paying for it

Note 1 to entry: This term can be used interchangeably with the term Credit mode when used in the context of operational modes.

Note 2 to entry: This term is usually used in conjunction with a system description whereas Credit mode is used when referring to the operational mode of a meter or account.

3.6.21
remote communications
transportation of a token or other message from a client to a server running a payment metering application process via some form of WAN and access network. This could be point to point, mesh radio, fibre optic connection, etc., and may travel through multiple devices and over multiple protocols before reaching the meter

3.6.22
repayable
credit_types such as emergency credit where an amount added to *current_credit_amount* of a "Credit" object has to be repaid before the Credit is selectable again

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.6.23
reserved credit
amount of credit that is held in reserve in the account of a payment meter, for use at a later time, at the discretion of the consumer

Note 1 to entry: The mechanism for reserving this credit may be subject to agreement between the utility supplier and the consumer. For example a proportion of every token may be added to the reserve Credit or the supplier might give the consumer an allowance every month, but these arrangements will be project specific.

3.6.24
selectable
specific state of a “Credit” object where the consumer’s immediate confirmation is needed before it can be brought into use

Note 1 to entry: For example, Emergency Credit has the nature of a short-term loan and should therefore only be deployed with the consumer’s agreement. The term refers respectively to the need to get agreement and to the fact of having received agreement. Only a ”Credit” made (1) Selectable can be (2) Selected / Invoked. Not all Credits need to be selected by an external trigger, as in most cases the meter application automatically performs this action.

3.6.25
selected/invoked
specific state of a “Credit” object where the value of *current_credit_amount* is included in the calculation of *available_credit* in the related “Account”

Note 1 to entry: This is the state of a Credit before becoming In use and is considered in the available_credit attribute of the “Account”, but is not yet being consumed by any Charge (due to a higher priority Credit being In use.

3.6.26
service
provision of a commodity (such as water, electricity gas or heat)

3.6.27
social credit
credit that is given free of payment for reasons such as the relief of poverty

Note 1 to entry: Typically such a credit is given at fixed times (e.g. monthly) in limited amounts. This particular type of credit could also be consumption based, such that that the consumer shall keep consumption below a limiting threshold in order to use the social credit. This could be controlled by the consumer being disconnected if the limit is breached.

Note 2 to entry: Social credits are modelled of “Credit” objects of type emergency_credit, time_based_credit, consumption_based_credit.

3.6.28
temporary debt
transient liability to the meter that accrues when *Charge*s are collected at a time when all credits are exhausted

Note 1 to entry: This temporary debt amount is accumulated in amount_to_clear in the “Account” object.

3.6.29
token
self-contained package of data related to the purchase of credit or to other system functions, embodied in a token carrier (q.v.). The token forms a link between source and destination of the transaction. The token contents may reflect money, energy, time, etc., in harmony with the currency declared in the meter

Note 1 to entry: Defined in IEC TR 62055-21:2005 as “<Equipment-related definition>[sic] information content including an instruction issued on a token carrier by a vending or management system that is capable of subsequent transfer to and acceptance by a specific payment meter, or one of a group of meters, with appropriate security”.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.6.30
token carrier
means of transferring a token from one system element to another, typically in material "physical" or electronic "virtual" form

Note 1 to entry: In a general sense, the token refers to the instruction and information being transferred, while the token carrier refers to the physical device being used to carry the instruction and information, or to the communications medium in the case of a virtual token carrier.

3.6.31
token carrier interface
interface between the token carrier and the payment metering installation

Note 1 to entry: For example, it may be a keypad for numeric tokens, or a physical token carrier acceptor, or a communications connection to a local or remote machine for a virtual token carrier interface.

Note 2 to entry: The token carrier interface may also be used to pass additional information to or from the payment meter, such as for the purposes of payment system management.

3.6.32
top-up
credit token
credit purchased by the consumer and capable of being delivered in the form of a token (as well as by other means) in a physical or virtual token carrier

3.6.33
transient device communications
transportation of a token within a payment metering installation through some electronic communication mechanism involving a transient device. This could be done by local radio, galvanic connection, optical connection, etc., from various devices, e.g. HHT, mobile phone

Note 1 to entry: In Home Displays are not classed as transient devices despite the fact that they may operate in a transient manner as far as the network is concerned. Transient devices shall be considered as devices that join the network for a short time and only very rarely. In home displays are considered to be on the network for long periods and only absent for very short periods in comparison to the life of the network.

3.6.34
vend
operation or transaction resulting in the available credit held on a payment meter to be increased by use of a credit token

Note 1 to entry: Vend would normally relate to a transaction in conjunction with a vending system at a point of sale, resulting in the creation of a token that can be transported by means of a physical or virtual token carrier.

3.7 Terms and definitions related to the Arbitrator IC (see 4.5.12)

3.7.1
action
operation that can be requested locally or remotely from the server

3.7.2
actor
entity requesting an action

Note 1 to entry: It can be the local application process or a client.

3.7.3
arbitrator
function modelled in COSEM that can determine, based on pre-configured rules, which action is carried out when multiple actors request potentially conflicting actions to control the same resource

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.8 Abbreviated terms

Abbreviation	Explanation
3GPP	3rd Generation Partnership Project
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Network
AA	Application Association
AARE	A-Associate Response – an APDU of the ACSE
AARQ	A-Associate Request – an APDU of the ACSE
ACSE	Association Control Service Element
ADP	Primary Station Address
ADS	Secondary Station Address
AGC	Automatic Gain Control
AL	Application layer
AP	Application process
APDU	Application Protocol Data Unit
APS	Application Support Sublayer (ZigBee® term)
ARFCN	Absolute radio-frequency channel number
ASE	Application Service Element
A-XDR	Adapted Extended Data Representation (IEC 61334-6)
base_name	The short_name corresponding to the first attribute (“logical_name”) of a COSEM object
BCD	Binary Coded Decimal
BER	Bit Error Rate
CBCP	CallBack Control Protocol (PPP)
CC	Current Credit (S-FSK PLC profile)
CDMA	Code Division Multiple Access
CENELEC	European Committee for Electrotechnical Standardization
CHAP	Challenge Handshake Authentication Protocol
CIASE	Configuration Initiation Application Service Element (S-FSK PLC profile)
class_id	Interface class identification code
CLI	Calling Line Identity
COSEM	Companion Specification for Energy Metering
COSEM object	An instance of a COSEM interface class
CPAS	Common Part Adaptation Sublayer
CRC	Cyclic Redundancy Check
CSD	Circuit Switched Data
CSAP	Client Service Access Point
CSMA	Carrier Sense Multiple Access
CtoS	Client to Server challenge
CU	Currently Unused
DC	Delta credit (S-FSK PLC profile)
DevAddr	(Related to LoRaWAN) A 32-bit non-unique identifier assigned to an end-device statically or dynamically after a Join Procedure (depending on the activation mode)
DHCP	Dynamic Host Configuration Protocol
DIB	Data Information Block (M-Bus)
DIF	Data Information Field (M-Bus)
DL	Data Link

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abbreviation	Explanation
DLMS®	Device Language Message Specification
DLMS® UA	DLMS User Association
DLPDU	Data Link Protocol Data Unit
DNS	Domain Name Server
DSCP	Differentiated Services Code Point
DSSID	Direct Switch ID
EAP	Extensible Authentication Protocol
eARFCN	Enhanced Absolute radio-frequency channel number
EDGE	Enhanced Data rates for GSM Evolution
EMC	Emergency Credit (in relation to payment metering)
ERP	Enterprise Resource Planning
EUI-48	48-bit Extended Unique Identifier
EUI-64	64-bit Extended Unique Identifier
E-UTRA	Evolved UMTS Terrestrial Radio Access
FCC	Federal Communications Commission
FFD	Full-Function Device
FIFO	First-In-First-Out
F/R	Fragmentation and Reassembly
FTP	File Transfer Protocol
GCM	Galois/Counter Mode, an algorithm for authenticated encryption with associated data
GMT	Greenwich Mean Time. Replaced by Coordinated Universal Time (UTC).
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HAN	Home Area Network
HART	Highway Addressable Remote Transducer see http://www.hartcomm.org/ (in relation with the Sensor manager interface class)
HDLC	High-level Data Link Control
HES	Head End System
HHT	Hand Held Terminal
HLS	High Level Security Authentication
HSDPA	High-Speed Downlink Packet Access
HS-PLC	High-Speed Power Line Carrier
IANA	Internet Assigned Numbers Authority
IB	Information Base
IC	Interface Class (COSEM)
IC	Initial credit (S-FSK PLC profile)
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPCP	Internet Protocol Control Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abbreviation	Explanation
ISP	Internet Service Provider
IT	Information Technology
ITU-T	International Telecommunication Union – Telecommunication
KEK	Key Encryption Key
LA	Local Area
LAC	Local Area Code
LAN	Local Area Network
LBD	(6)LoWPAN Bootstrapping Device
LCID	Local Connection Identifier
LCP	Link Control Protocol
LDN	Logical Device Name
LLC	Logical Link Control (sublayer)
LLS	Low Level Security
LN	Logical Name
LNID	Local Node Identifier
LOADng	6LoWPAN Ad Hoc On-Demand Distance Vector Routing Next Generation (LOADng)
LQI	Link Quality Indicator (ZigBee ® term)
LSB	Least Significant Bit
LSID	Local Switch Identifier
LTE	Long Term Evolution (Wireless communication)
M	mandatory
M2M	Machine to Machine
MAC	Medium Access Control
M-Bus	Meter Bus
MCC	Mobile Country Code
MD5	Message Digest Algorithm 5
MIB	Management Information Base (S-FSK PLC profile)
MID	Measuring Instruments Directive 2004/22/EC of the European Parliament and of the Council
MMO	Matyas-Meyer-Oseas hash (ZigBee ® term)
MNC	Mobile Network Code
MPAN	(UK term) Meter Point Access Number – reference of the location of the Electricity meter on the electricity distribution network.
MPDU	MAC Protocol Data Unit
MSB	Most Significant Bit
MSDU	MAC Service Data Unit
MT	Mobile Termination
NB	Narrow-band
ND	Neighbour Discovery
NRSRP	Narrowband Reference Signal Received Power
NRSRQ	Narrowband Reference Signal Received Quality
NTP	Network Time Protocol
O	optional
OBIS	OBject Identification System
OFDM	Orthogonal Frequency Division Multiplexing

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abbreviation	Explanation
OTA	Over the Air – Refers to Firmware Upgrade using ZigBee ®
PAN	Personal Area Network (Term used in relation to G3-PLC [1]) and ZigBee ®
Pad	Padding
PAP	Password Authentication Protocol
PCLK	Pre-Configured Link Key (ZigBee® term)
PDU	Protocol Data Unit
PhL, PHY	Physical Layer
PIB	PLC Information Base
PID	Program Identifier
PIN	Personal Identity Number
PLC	Power Line Carrier
PLMN	Public Land Mobile Network
PNPDU	Promotion Needed PDU
POS	Point Of Sale (Payment metering)
POS	Personal Operating Space (ZigBee ®)
PPDU	Physical Protocol Data Unit
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RB	Radio Band
REJ PDU	Reject Protocol Data Unit
RFC	Request for Comments; a document published by the Internet Engineering Task Force
RFD	Reduced Function Device
ROHC	Robust Header Compression
RREP	Route Reply
RREQ	Route Request
RRER	Route Error
RSRQ	Reference Signal Received Quality
RSRP	Reference Signal Received Power
RSSI	Received Signal Strength Indication (ZigBee® term)
SAP	Service Access Point
SAS	Startup Attribute Set (ZigBee® term)
SCP	Shared Contention Period
SDU	Service Data Unit
SE	Smart Energy
SEP	Smart Energy Profile (ZigBee® term)
S-FSK	Spread – Frequency Shift Keying
SGERG88	Method for calculation of compressibility (Gas metering)
SHA	Secure Hash Algorithm
SID	Switch identifier
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
SN	Short Name
SNA	Subnetwork Address

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abbreviation	Explanation
SSAS	Service Specific Adaptation Sublayer
SSCS	Service Specific Convergence Layer
StoC	Server to Client Challenge
TAB	In the case of the EURIDIS profiles without DLMS® and without DLMS®/COSEM: data code. In the case of profiles using DLMS® or DLMS®/COSEM: value at which the equipment is programmed for Discovery
TABi	List of TAB field
TCC	Transmission Control Code (IPv4)
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TOU	Time of use
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UNC	Unconfigured (S-FSK PLC profile)
UTC	Coordinated Universal Time
VIB	Value Information Block (M-Bus)
VIF	Value Information Field (M-Bus)
VZ	Billing period counter (Form *Vorwertzähler* in German, see DIN 43863-3)
wake-up	trigger the meter to connect to the communication network to be available to a client (e.g. HES)
WAN	Wide Area Network
wM-Bus	Wireless M-Bus
ZTC	ZigBee® Trust Center
[1)] In the case of the G3-PLC technology, PAN may be defined as PLC Area Network.	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4 The COSEM interface classes

4.1 Basic principles

4.1.1 General

This subclause 4.1 describes the basic principles on which the COSEM interface classes (ICs) are built. It also gives a short overview on how interface objects – instantiations of the ICs – are used for communication purposes. Data collection systems and metering equipment from different vendors, following these specifications, can exchange data in an interoperable way.

For specification purposes, this standard uses the technique of object modelling.

An object is a collection of attributes and methods. Attributes represent the characteristics of an object. The value of an attribute may affect the behaviour of an object. The first attribute of any object is the *logical_name*. It is one part of the identification of the object. An object may offer a number of methods to either examine or modify the values of the attributes.

Objects that share common characteristics are generalized as an interface class, identified with a class_id. Within a specific IC, the common characteristics (attributes and methods) are described once for all objects. Instantiations of ICs are called COSEM interface objects.

Manufacturers may add proprietary methods and attributes to any object; see 4.1.2.

Figure 2 illustrates these terms by means of an example:

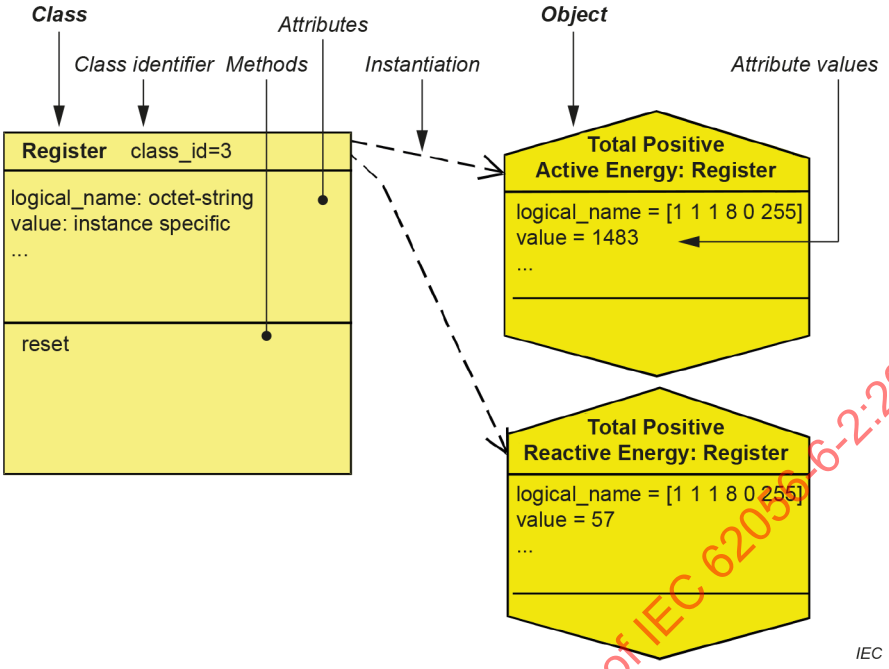


Figure 2 – An interface class and its instances

The IC "Register" is formed by combining the features necessary to model the behaviour of a generic register (containing measured or static information) as seen from the client (data collection system, hand held terminal). The contents of the register are identified by the attribute *logical_name*. The *logical_name* contains an OBIS identifier (see IEC 62056-6-1:2023. The actual (dynamic) content of the register is carried by its *value* attribute.

Defining a specific meter means defining several specific objects. In the example of Figure 2, the meter contains two registers; i.e. two specific instances of the IC "Register" are instantiated. Through the instantiation, one COSEM object becomes a "total, positive, active energy register" whereas the other becomes a "total, positive, reactive energy register".

NOTE The COSEM interface objects (instances of COSEM ICs) represent the behaviour of the meter as seen from the "outside". Therefore, modifying the value of an attribute – for example resetting the *value* attribute of a register – is always initiated from the outside. Internally initiated changes of the attributes – for example updating the *value* attribute of a register – are not described in this model.

4.1.2 Referencing methods

Attributes and methods of COSEM objects can be referenced in two different ways:

Using logical names (LN referencing): In this case, the attributes and methods are referenced via the identifier of the COSEM object instance to which they belong.

The reference for an attribute is: class_id, value of the *logical_name* attribute, attribute_index.

The reference for a method is: class_id, value of the *logical_name* attribute, method_index, where:

- attribute_index is used as the identifier of the attribute required. Attribute indexes are specified in the definition of each IC. They are positive numbers starting with 1. Proprietary attributes may be added: these shall be identified with negative numbers;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- method_index is used as the identifier of the method required. Method indexes are specified in the definition of each IC. They are positive numbers starting with 1. Proprietary methods may be added: these shall be identified with negative numbers.

Using short names (SN referencing): This kind of referencing is intended for use in simple devices. In this case, each attribute and method of a COSEM object is identified with a 13-bit integer. The syntax for the short name is the same as the syntax of the name of a DLMS® named variable. See IEC 61334-4-41:1996 and IEC 62056-5-3:2023 Clause 8.

4.1.3 Reserved base_names for special COSEM objects

In order to facilitate access to devices using SN referencing, some short_names are reserved as base_names for special COSEM objects. The range for reserved base_names is from 0xFA00 to 0xFFF8. The following specific base_names are defined, see Table 1.

Table 1 – Reserved base_names for SN referencing

Base_name (objectName)	COSEM object
0xFA00	Association SN
0xFB00	Script table (instantiation: Broadcast “Script table”)
0xFC00	SAP assignment
0xFD00	“Data” or “Register” object containing the “COSEM logical device name” in the attribute "value"

4.1.4 Class description notation

4.1.4.1 General

This subclause describes the notation used to define the ICs.

A short text describes the functionality and application of the IC. Table 2 gives an overview of the IC including the class name, the attributes, and the methods. Each attribute and method shall be described in detail. The template is shown below.

Table 2 – Interface class overview

Class name		**Cardinality**	**class_id, version**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. …	(…)	…				x + 0x…
3. …	(…)	…				x + 0x…
Specific methods (if required)		***m/o***				
1.		…				x + 0x…
2.		…				x + 0x…
3.		…				x + 0x…

4.1.4.2 Class name

Describes the interface class (e.g. “Register”, “Clock”, “Profile generic”…).

NOTE Interface classes names are mentioned in quotation marks.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.1.4.3 Cardinality

Specifies the number of instances of the IC within a logical device (see 4.1.8).

value The IC shall be instantiated exactly “value” times.

min...max. The IC shall be instantiated at least “min.” times and at most “max.” times. If min. is zero (0) then the IC is optional, otherwise (min. > 0) "min." instantiations of the IC are mandatory.

4.1.4.4 class_id

Identification code of the IC (range 0 to 65 535). The class_id of each object is retrieved together with the logical name by reading the object_list attribute of an “Association LN” / ”Association SN” object.

– class_id-s from 0 to 8 191 are reserved to be specified by the DLMS® UA.
– class_id-s from 8 192 to 32 767 are reserved for manufacturer specific ICs.
– class_id-s from 32 768 to 65 535 are reserved for user group specific ICs.

The DLMS® UA reserves the right to assign ranges to individual manufacturers or user groups.

class_id-s 32768 to 32778 are allocated to the STS Association.

4.1.4.5 version

Identification code of the version of the IC. The version of each object is retrieved together with the class_id and the logical name by reading the object_list attribute of an “Association LN” / ”Association SN” object.

Within one logical device, all instances of a certain IC shall be of the same version.

4.1.4.6 Attributes

Specifies the attributes that belong to the IC.

(dyn.) Classifies an attribute that carries a process value, which is updated by the meter itself.

(static) Classifies an attribute, which is not updated by the meter itself (for example configuration data).

NOTE 1 There are some attributes which may be either static or dynamic depending on the application. In these cases this property is not indicated.

NOTE 2 Attribute names use the underscore notation. When mentioned in the text they are in *italic*. Example: *logical_name*.

4.1.4.7 logical_name

octet-string It is always the first attribute of an IC. It identifies the instantiation (COSEM object) of this IC. The value of the *logical_name* conforms to OBIS; see Clauses 6 and IEC 62056-6-1:2023.

4.1.4.8 Data type

Defines the data type of an attribute; see 4.1.5.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.1.4.9 Min.

Specifies if the attribute has a minimum value.

x The attribute has a minimum value.

<*empty*> The attribute has no minimum value.

4.1.4.10 Max.

Defines if the attribute has a maximum value.

x The attribute has a maximum value.

<*empty*> The attribute has no maximum value.

4.1.4.11 Def.

Specifies if the attribute has a default value. This is the value of the attribute after reset.

x The attribute has a default value.

<*empty*> The default value is not defined by the IC specification.

4.1.4.12 Short name

When Short Name (SN) referencing is used, each attribute and method of object instances has to be mapped to short names.

The base_name x of each object instance is the DLMS® named variable the logical name attribute is mapped to. It is selected in the implementation phase. The IC definition specifies the offsets for the other attributes and for the methods.

4.1.4.13 Specific methods

Provides a list of the specific methods that belong to the object.

Method Name () The method has to be described in the subsection "Method description".

NOTE Method names use the underscore notation. When mentioned in the text they are in *italic*. Example: *add_object*.

4.1.4.14 m/o

Defines if the method is mandatory or optional.

m (mandatory) The method is mandatory.

o (optional) The method is optional.

4.1.4.15 Attribute description

Describes each attribute with its data type (if the data type is not simple), its data format and its properties (minimum, maximum and default values).

4.1.4.16 Method description

Describes each method and the invoked behaviour of the COSEM object(s) instantiated.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE Services for accessing attributes or methods by the protocol are specified in IEC 62056-5-3:2023, 6.6 to 6.17.

4.1.4.17 Selective access

The xDLMS attribute-related services typically reference the entire attribute. However, for certain attributes selective access to just a part of the attribute may be provided. The part of the attribute is identified by specific selective access parameters. These are defined as part of the attribute specification.

Selective access is available with the following interface class attributes and methods:

- "Profile generic" objects, **buffer** attribute;
- "Association SN" objects, **object_list** and **access_rights_list** attribute;
- "Association LN" objects, **object_list** attribute;
- "Compact data", objects, **compact_buffer** attribute;
- "Push" objects, **push_object_list** attribute;
- "Data protection" objects, **protection_object_list** attribute **get_protected_attributes** method and **set_protected_attributes** method.

4.1.5 Common data types

Table 3 contains the list of data types usable for attributes of COSEM objects.

Table 3 – Common data types

Type description	Tag [a]	Definition	Value range
-- simple data types			
null-data	[0]		
boolean	[3]	boolean	TRUE or FALSE
bit-string	[4]	An ordered sequence of boolean values	
double-long	[5]	Integer32	-2 147 483 648... 2 147 483 647
double-long-unsigned	[6]	Unsigned32	0...4 294 967 295
	[7]	Tag of the "floating-point" type in IEC 61334-4-41:1996, not usable in DLMS®/COSEM. See tags [23] and [24]	
octet-string	[9]	An ordered sequence of octets (8 bit bytes)	
visible-string	[10]	An ordered sequence of ASCII characters	
	[11]	Tag of the "time" type in IEC 61334-4-41:1996, not usable in DLMS®/COSEM. See tag [27]	
utf8-string	[12]	An ordered sequence of characters encoded as UTF-8	
bcd	[13]	binary coded decimal	
integer	[15]	Integer8	-128...127
long	[16]	Integer16	-32 768...32 767
unsigned	[17]	Unsigned8	0...255
long-unsigned	[18]	Unsigned16	0...65 535
long64	[20]	Integer64	$-2^{63}...2^{63}-1$
long64-unsigned	[21]	Unsigned64	$0...2^{64}-1$
enum	[22]	The elements of the enumeration type are defined in the *Attribute description* or *Method description* section of a COSEM IC specification.	0...255
float32	[23]	OCTET STRING (SIZE(4))	For formatting,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Type description	Tag [a]	Definition	Value range
float64	[24]	OCTET STRING (SIZE(8))	see 4.1.6.2.
date-time	[25]	OCTET STRING SIZE(12))	For formatting, see 4.1.6.1.
date	[26]	OCTET STRING (SIZE(5))	
time	[27]	OCTET STRING (SIZE(4))	
delta-integer	[28]	Integer8	-128…127
delta-long	[29]	Integer16	-32 768…32 767
delta-double-long	[30]	Integer32	-2 147 483 648… 2 147 483 647
delta-unsigned	[31]	Unsigned8	0…255
delta-long-unsigned	[32]	Unsigned16	0…65 535
delta-double-long-unsigned	[33]	Unsigned32	0…4 294 967 295
-- complex data types			
array	[1]	The elements of the array are defined in the *Attribute* or *Method description* section of a COSEM IC specification.	
structure	[2]	The elements of the structure are defined in the *Attribute or Method description* section of a COSEM IC specification.	
compact array	[19]	Provides an alternative, compact encoding of complex data.	
-- CHOICE		For some COSEM interface objects attributes, the data type may be chosen at instantiation, in the implementation phase of the COSEM server. The server always shall send back the data type and the value of each attribute, so that together with the logical name an unambiguous interpretation is ensured. The list of possible data types is defined in the “Attribute description” section of a COSEM IC specification.	

[a] The tags are as defined in IEC 62056-5-3:2023, Clause 8.

4.1.6 Data formats

4.1.6.1 Date and time formats

Date and time information may be represented using the data type *octet-string*.

NOTE 1 In this case the encoding includes the tag of the data type *octet-string*, the length of the octet-string and the elements of *date*, *time* and /or *date-time* as applicable.

Date and time information may be also represented using the data types *date, time* and *date-time*.

NOTE 2 In these cases, the encoding includes only the tag of the data types *date*, *time* or *date-time* as applicable and the elements of date, time or date-time.

NOTE 3 The (SIZE ( )) specifications are applicable only when date, time or date time are represented by the data types *date*, *time* or *date-time*.

date OCTET STRING (SIZE(5))

{
year highbyte,
year lowbyte,
month,
day of month,
day of week
}

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

year: interpreted as long-unsigned
range 0…big
0xFFFF = not specified

year highbyte and year lowbyte represent the 2 bytes of the long-unsigned

month: interpreted as unsigned
range 1…12, 0xFD, 0xFE, 0xFF
1 is January
0xFD = daylight_savings_end
0xFE = daylight_savings_begin
0xFF = not specified

dayOfMonth: interpreted as unsigned
range 1…31, 0xFD, 0xFE, 0xFF
0xFD = 2nd last day of month
0xFE = last day of month
0xE0 to 0xFC = reserved
0xFF = not specified

dayOfWeek: interpreted as unsigned
range 1…7, 0xFF
1 is Monday
0xFF = not specified

For repetitive dates, the unused parts shall be set to "not specified".

For countries not using the Gregorian calendar, Month 1 is the starting month of the calendar and the range of dayOfMonth may be different.

The elements dayOfMonth and dayOfWeek shall be interpreted together:

– if last dayOfMonth is specified (0xFE) and dayOfWeek is wildcard, this specifies the last calendar day of the month;
– if last dayOfMonth is specified (0xFE) and an explicit dayOfWeek is specified (for example 7, Sunday) then it is the last occurrence of the weekday specified in the month, i.e. the last Sunday;
– if the year is not specified (0xFFFF), and dayOfMonth and dayOfWeek are both explicitly specified, this shall be interpreted as the dayOfWeek on, or following dayOfMonth;
– if the year and month are specified, and both the dayOfMonth and dayOfWeek are explicitly specified but the values are not consistent it shall be considered as an error.

Examples:

1) year = 0xFFFF, month = 0x FF, dayOfMonth = 0xFE, dayofWeek = 0xFF: last day of the month in every year and month;

2) year = 0xFFFF, month = 0x FF, dayOfMonth = 0xFE, dayofWeek = 0x07: last Sunday in every year and month;

3) year = 0xFFFF, month = 0x03, dayOfMonth = 0xFE, dayofWeek = 0x07: last Sunday in March in every year;

4) year = 0xFFFF, month = 0x03, dayOfMonth = 0x01, dayofWeek = 0x07: first Sunday in March in every year;

5) year = 0xFFFF, month = 0x03, dayOfMonth = 0x16, dayofWeek = 0x05: fourth Friday in March in every year;

6) year = 0xFFFF, month = 0x0A, dayOfMonth = 0x16, dayofWeek = 0x07: fourth Sunday in October in every year;

7) year = 0x07DE, month = 0x08, dayOfMonth = 0x13, (2014.08.13, Wednesday) dayofWeek = 0x02 (Tuesday): error, as the dayOfMonth and dayOfWeek in the given year and month do not match.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

time OCTET STRING (SIZE(4))

{
hour,
minute,
second,
hundredths
}

Where:

hour: interpreted as unsigned
range 0…23, 0xFF,

minute: interpreted as unsigned
range 0…59, 0xFF,

second: interpreted as unsigned
range 0…59, 0xFF,

hundredths: interpreted as unsigned
range 0…99, 0xFF

For hour, minute, second and hundredths: 0xFF = not specified.

For repetitive times the unused parts shall be set to “not specified”.

date-time OCTET STRING (SIZE(12))
{
year highbyte,
year lowbyte,
month,
day of month,
day of week,
hour,
minute,
second,
hundredths of second,
deviation highbyte,
deviation lowbyte,
clock status
}

The elements of *date* and *time* are encoded as defined above. Some may be set to “not specified” as defined above.

In addition:

deviation: interpreted as long
range -720…+720 in minutes of local time to UTC
0x8000 = not specified

Deviation highbyte and deviation lowbyte represent the 2 bytes of the long.

clock_status interpreted as unsigned.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The bits are defined as follows:

bit 0 (LSB): invalid [a] value,
bit 1: doubtful [b] value,
bit 2: different clock base [c],
bit 3: invalid clock status [d],
bit 4: reserved,
bit 5: reserved,
bit 6: reserved,
bit 7 (MSB): daylight saving active [e]

0xFF = not specified

[a] Time could not be recovered after an incident. Detailed conditions are manufacturer specific (for example after the power to the clock has been interrupted). For a valid status, bit 0 shall not be set if bit 1 is set.

[b] Time could be recovered after an incident but the value cannot be guaranteed. Detailed conditions are manufacturer specific. For a valid status, bit 1 shall not be set if bit 0 is set.

[c] Bit is set if the basic timing information for the clock at the actual moment is taken from a timing source different from the source specified in clock_base.

[d] This bit indicates that at least one bit of the clock status is invalid. Some bits may be correct. The exact meaning shall be explained in the manufacturer's documentation.

[e] Flag set to true: the transmitted time contains the daylight saving deviation (summer time).

Flag set to false: the transmitted time does not contain daylight saving deviation (normal time).

4.1.6.2 Floating point number formats

Floating point number formats are defined in ISO/IEC 60559:2020.

The single format is:

1	8	23	…widths
s	*e*	*F*	
msb *lsb*		*msb* *lsb*	*…order*

where:

s is the sign bit;

e is the exponent; it is 8 bits wide and the exponent bias is +127;

f is the fraction, it is 23 bits.

With this, the value is (if 0 < e < 255):

$$\upsilon = (-1)^{s} \times 2^{e-127} \times (1 \times f)$$

The double format is:

1	11	52	…widths
s	*e*	*F*	
msb *lsb*		*msb* *lsb*	*…order*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

where:

s is the sign bit;

e is the exponent; it is 11 bits wide and the exponent bias is +1 023;

f is the fraction, it is 52 bits.

With this, the value is (if 0 < e < 2 047):

$$\upsilon = (-1)^{s} \times 2^{e-1023} \times (1 \times f)$$

For details, see ISO/IEC 60559:2020.

Floating-point numbers shall be represented as a fixed length octet-string, containing the 4 bytes (float32) of the single format or the 8 bytes (float64) of the double format floating-point number as specified above, most significant byte first.

EXAMPLE 1 The decimal value “1” represented in single floating-point format is:

Bit 31 Sign bit **0** 0: + 1: -	Bits 30-23 Exponent field: **01111111** Decimal value of exponent field and exponent: **127**-127 = 0	Bits 22-0 Significand 1.**00000000000000000000000** Decimal value of the significand: 1.0000000

NOTE The significand is the binary number 1 followed by the radix point followed by the binary bits of the fraction.

The encoding, including the tag of the data type is (all values are hexadecimal): 17 3F 80 00 00.

EXAMPLE 2 The decimal value “1” represented in double floating-point format is:

Bit 63 Sign bit **0** 0: + 1: -	Bits 62-52 Exponent field: **01111111111** Decimal value of exponent field and exponent: **1023**-1023 = 0	Bits 51-0 Significand 1.**0000000000000000000000000000000000000000000000000000** Decimal value of the significand: 1.0000000000000000

The encoding, including the tag of the data type is (all values are hexadecimal): 18 3F F0 00 00 00 00 00 00.

EXAMPLE 3 The decimal value “62056” represented in single floating-point format is:

Bit 31 Sign bit **0** 0: + 1: -	Bits 30-23 Exponent field: **10001110** Decimal value of exponent field and exponent: **142**-127 = 15	Bits 22-0 Significand 1.**11100100110100000000000** Decimal value of the significand: 1.8937988

The encoding, including the tag of the data type is (all values are hexadecimal): 17 47 72 68 00.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

EXAMPLE 4 The decimal value “62056” represented in double floating-point format is:

Bit 63	Bits 62-52	Bits 51-0
Sign bit	Exponent field: **10000001110**	Significand
0 0: + 1: -	Decimal value of exponent field and exponent: **1038**-1023 = 15	1.**11100100110100000000000000000000000000000000000000000** Decimal value of the significand: 1.8937988281250000

The encoding, including the tag of the data type is (all values are hexadecimal): 18 40 EE 4D 00 00 00 00 00.

4.1.6.3 Null- data and delta-value encoding

Null-data encoding and delta-value encoding are two methods to improve efficiency when array or arrays of structures have to be transferred, for example Profile generic IC buffer attribute or Register table IC table_cell_values attribute.

With null-data encoding, the value may be replaced by “null-data” if it can be unambiguously recovered from the previous value (for example for time: if it can be calculated from the previous value and capture_period; or for a value: if it is equal to the previous value).

With delta-value encoding the original value may be replaced by a delta-value as follows:

- the first value shall be encoded using an appropriate integer or unsigned data type;
- subsequent values shall be encoded the same way, or they may be replaced by a delta value encoded using an appropriate delta data type.

The two methods can be used individually or combined and they can be used with compression to further improve efficiency.

4.1.7 The COSEM server model

The COSEM server is structured into three hierarchical levels as shown in Figure 3:

Level 1: Physical device
Level 2: Logical device
Level 3: Accessible COSEM objects

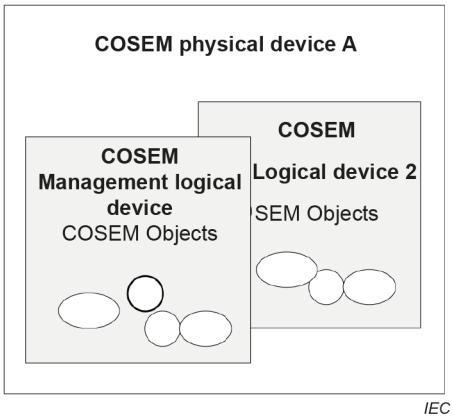


Figure 3 – The COSEM server model

The example in Figure 4 shows how a combined metering device can be structured using the COSEM server model.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

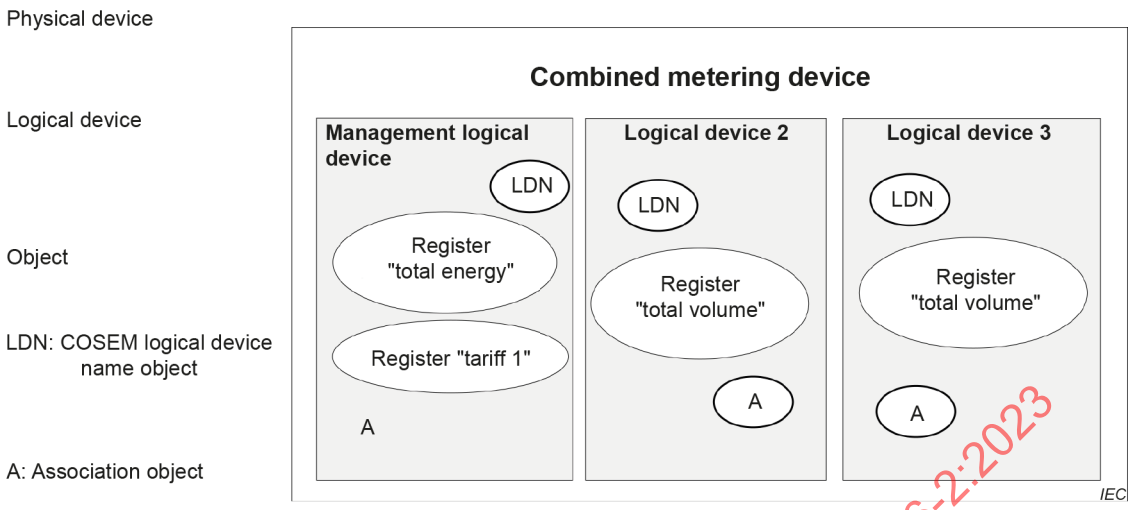


Figure 4 – Combined metering device

4.1.8 The COSEM logical device

4.1.8.1 General

The COSEM logical device contains a set of COSEM objects. Each physical device shall contain a “Management logical device”.

The addressing of COSEM logical devices shall be provided by the addressing scheme of the lower layers of the protocol stack used.

4.1.8.2 COSEM logical device name (LDN)

The COSEM logical device can be identified by its unique COSEM LDN. This name can be retrieved from an instance of IC “SAP assignment” (see 4.4.5), or from a COSEM object “COSEM logical device name” (see 6.2.35).

The LDN is defined as an octet-string of up to 16 octets. The first three octets shall carry the manufacturer identifier [2]. The manufacturer shall ensure that the LDN, starting with the three octets identifying the manufacturer and followed by up to 13 octets, is unique.

4.1.8.3 The “association view” of the logical device

In order to access COSEM objects in the server, an application association (AA) shall first be established with a client. AAs identify the partners and characterize the context within which the associated applications will communicate. The major parts of this context are:

- the application context;
- the authentication mechanism;
- the xDLMS context.

AAs are modelled by special COSEM objects:

- instances of the IC “Association SN” – see 4.4.3 – are used with short name referencing;
- instances of the IC “Association LN” – see 4.4.4 – are used with logical name referencing.

---

2 Administered by the DLMS® User Association, in cooperation with the FLAG Association.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Depending on the AA established between the client and the server, different access rights may be granted by the server. Access rights concern a set of COSEM objects – the visible objects – that can be accessed ('seen') within the given AA. In addition, access to attributes and methods of these COSEM objects may also be restricted within the AA (for example a certain type of client can only read a particular attribute of a COSEM object, but cannot write it). Access right may also stipulate required cryptographic protection.

The list of the visible COSEM objects – the "association view" – can be obtained by the client by reading the *object_list* attribute of the appropriate association object.

4.1.8.4 Mandatory contents of a COSEM logical device

The following objects shall be present in each COSEM logical device. They shall be accessible for GET/Read in all AAs with this logical device:

- COSEM logical device name object;
- current "Association" (LN or SN) object.

If the "SAP Assignment" object is present, then the COSEM logical device name object does not have to be present.

For identifying the firmware the following objects are mandatory:

- an active firmware identifier object that holds the identifier of the currently active firmware;
- an active firmware signature object that holds the digital signature of the currently active firmware.

NOTE The digital signature algorithm is not specified here.

If a Logical Device has multiple firmwares then an active firmware identifier object and an active firmware signature object shall be present for each firmware.

The following objects may be optionally present:

- one or more active firmware version object(s) that hold(s) the version of the currently active firmware.

4.1.8.5 Management logical device

As specified in 4.1.8.1, the management logical device is a mandatory element of any physical device. It has a reserved address. It shall support an AA to a public client with the lowest level security (no security) authentication. Its role is to support revealing the internal structure of the physical device and to support notification of events in the server.

In addition to the "Association" object modelling the AA with the public client, the management logical device shall contain a "SAP assignment" object, giving its SAP and the SAP of all other logical devices within the physical device. The SAP assignment object shall be readable at least by the public client.

If there is only one logical device within the physical device, the "SAP assignment" object may be omitted.

4.1.9 Information security

DLMS®/COSEM provides several information security features for accessing and transporting data:

- data access security controls access to the data held by a DLMS®/COSEM server;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- data transport security allows the sending party to apply cryptographic protection to the xDLMS APDUs sent. This requires ciphered APDUs. The receiving party can remove or check this protection;
- COSEM data security allows protecting COSEM attribute values, as well as method invocation and return parameters.

For a description of these security mechanisms, see IEC 62056-5-3:2023 ,Clause 5.

Information security is provided on the DLMS®/COSEM Application layer level and on the COSEM object level and it is supported / managed by the following objects:

- “Association SN”, see 4.4.3;
- “Association LN”, see 4.4.4; and
- “Security setup”, see 4.4.7;
- “Data protection”, see 4.4.9.

4.2 Overview of the COSEM interface classes

The ICs defined currently and the relations between them are shown in Figure 5, Figure 6 and Figure 7.

NOTE 1 The IC “base” itself is not specified explicitly. It contains only one attribute *logical_name*.

NOTE 2 In the description of the "Demand register", “Clock” and “Profile generic” ICs, the 2nd attributes are labelled differently from that of the 2nd attribute of the “Data” IC, namely *current_average_value*, *time* and *buffer* vs. *value*. This is to emphasize the specific nature of the *value*.

NOTE 3 In these Figures the interface classes are presented in each group by increasing class_id. In the clauses specifying the various groups of interface classes, the new interface classes are put at the end of the relevant clause.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

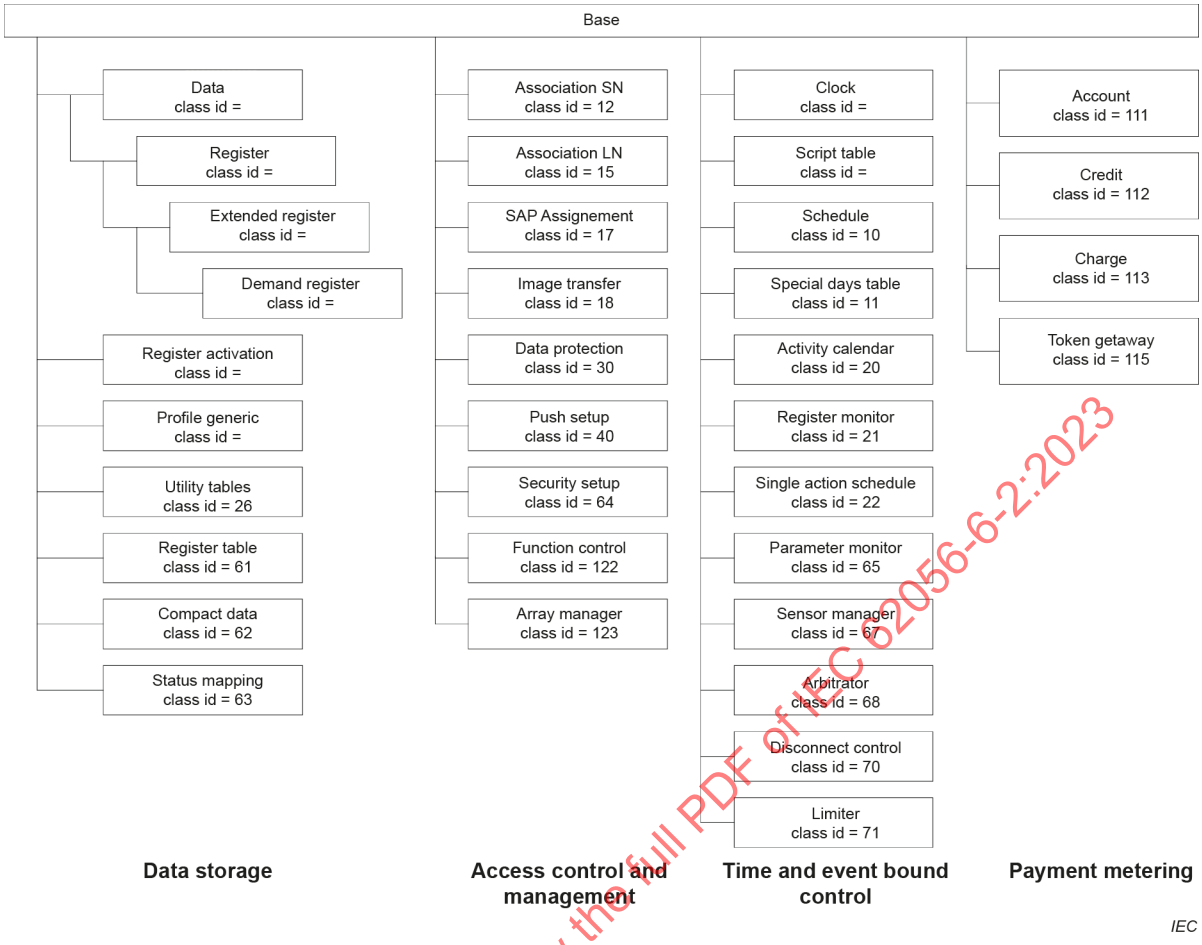
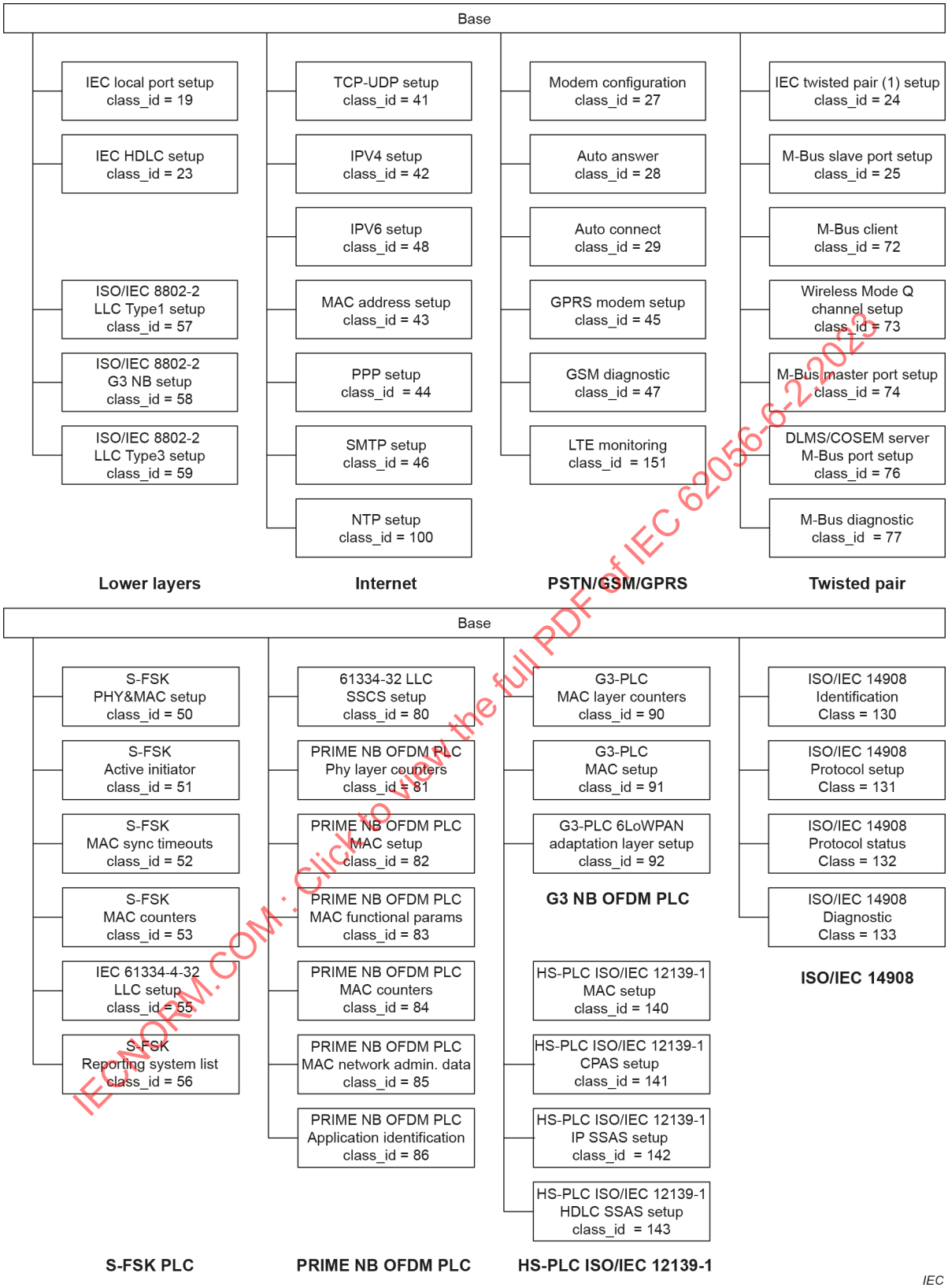


Figure 5 – Overview of the interface classes – Part 1

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 6 – Overview of the interface classes – Part 2

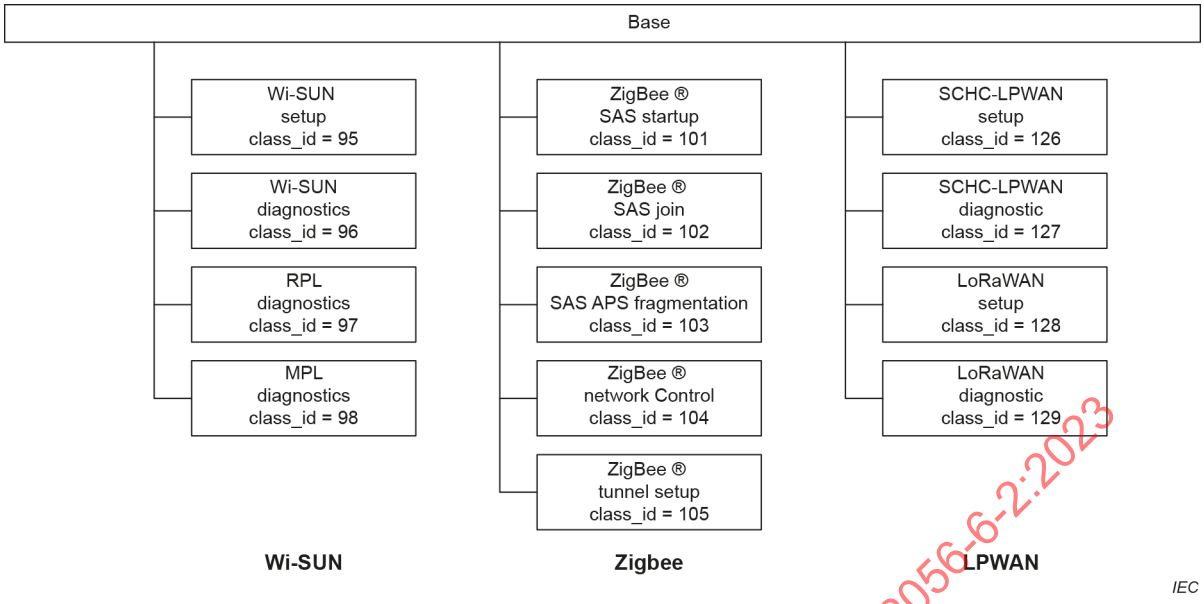


Figure 7 – Overview of the interface classes – Part 3

Table 4 lists the interface classes by class_id.

Table 4 – List of interface classes by class_id

Interface class name	class_id	version(s)	Subclause
Data	1	0	4.3.1
Register	3	0	4.3.2
Extended register	4	0	4.3.3
Demand register	5	0	4.3.4
Register activation	6	0	4.3.5
Profile generic	7	1	4.3.6
		0	5.3.1
Clock	8	0	4.5.1
Script table	9	0	4.5.2
Schedule	10	0	4.5.3
Special days table	11	0	4.5.4
Association SN	12	4	4.4.3
		3	5.4.4
		2	5.4.3
		1	5.4.2
		0	5.4.1
Association LN	15	3	4.4.4
		2	5.4.7
		1	5.4.6
		0	5.4.5
SAP Assignment	17	0	4.4.5
Image transfer	18	0	4.4.6
IEC local port setup	19	1	4.7.1
		0	5.7.1

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Interface class name	class_id	version(s)	Subclause
Activity calendar	20	0	4.5.5
Register monitor	21	0	4.5.6
Single action schedule	22	0	4.5.7
IEC HDLC setup	23	1 0	4.7.2 5.7.2
IEC twisted pair (1) setup	24	1 0	4.7.3 5.7.3
M-BUS slave port setup	25	0	4.8.2
Utility tables	26	0	4.3.7
Modem configuration PSTN modem configuration	27	1 0	4.7.4 5.7.4
Auto answer	28	2 1 0	4.7.5 – 5.7.5
Auto connect PSTN Auto dial	29	2 1 0	4.7.6 5.7.7 5.7.6
Data protection	30	0	4.4.9
Push setup	40	2 1 0	4.4.8.2 5.4.10 5.4.9
TCP-UDP setup	41	0	4.9.1
IPv4 setup	42	0	4.9.2
MAC address setup (Ethernet setup)	43	0	4.9.4 4.12.10
PPP setup	44	0	4.9.5
GPRS modem setup	45	0	4.7.7
SMTP setup	46	0	4.9.6
GSM diagnostic	47	2 1 0	4.7.8 5.7.9 5.7.8
IPv6 setup	48	0	4.9.3
S-FSK Phy&MAC setup	50	1 0	4.10.3 5.10.1
S-FSK Active initiator	51	0	4.10.4
S-FSK MAC synchronization timeouts	52	0	4.10.5
S-FSK MAC counters	53	0	4.10.6
IEC 61334-4-32 LLC setup S-FSK IEC 61334-4-32 LLC setup	55	1 0	4.10.7 5.10.2
S-FSK Reporting system list	56	0	4.10.8
ISO/IEC 8802-2 LLC Type 1 setup	57	0	4.11.2
ISO/IEC 8802-2 LLC Type 2 setup	58	0	4.11.3
ISO/IEC 8802-2 LLC Type 3 setup	59	0	4.11.4
Register table	61	0	4.3.8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Interface class name	class_id	version(s)	Subclause
Compact data	62	1	4.3.10
		0	5.3.2
Status mapping	63	0	4.3.9
Security setup	64	1	4.4.7
		0	5.4.8
Parameter monitor	65	1	4.5.10
		0	5.5.1
Sensor manager	67	0	4.5.11
Arbitrator	68	0	4.5.12
Disconnect control	70	0	4.5.8
Limiter	71	0	4.5.9
M-Bus client	72	1	4.8.3
		0	5.8.1
Wireless Mode Q channel	73	0	4.8.4
M-Bus master port setup	74	0	4.8.5
DLMS®/COSEM server M-Bus port setup	76	0	4.8.6
M-Bus diagnostic	77	0	4.8.7
61334-4-32 LLC SSCS setup	80	0	4.12.3
PRIME NB OFDM PLC Physical layer counters	81	0	4.12.5
PRIME NB OFDM PLC MAC setup	82	0	4.12.6
PRIME NB OFDM PLC MAC functional parameters	83	0	4.12.7
PRIME NB OFDM PLC MAC counters	84	0	4.12.8
PRIME NB OFDM PLC MAC network administration data	85	0	4.12.9
PRIME NB OFDM PLC Application identification	86	0	4.12.11
G3-PLC MAC layer counters	90	1	4.13.3
G3 NB OFDM PLC MAC layer counters		0	5.13.3
G3-PLC MAC setup	91	2	4.13.4
G3-PLC MAC setup		1	5.13.6
G3 NB OFDM PLC MAC setup		0	5.13.3
G3-PLC 6LoWPAN adaptation layer setup	92	2	4.13.5
G3-PLC 6LoWPAN adaptation layer setup		1	5.13.7
G3 NB OFDM PLC 6LoWPAN adaptation layer setup		0	5.13.4
Wi-SUN setup	95	0	4.17.1
Wi-SUN diagnostic	96	0	4.17.2
RPL diagnostic	97	0	4.17.3
MPL diagnostic	98	0	4.17.4
NTP Setup	100	0	4.9.7
ZigBee® SAS startup	101	0	4.15.2
ZigBee® SAS join	102	0	4.15.3
ZigBee® SAS APS fragmentation	103	0	4.15.4
ZigBee® network control	104	0	4.15.5
ZigBee® tunnel setup	105	0	4.15.6
Account	111	0	4.6.2
Credit	112	0	4.6.3

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Interface class name	class_id	version(s)	Subclause
Charge	113	0	4.6.4
Token gateway	115	0	4.6.5
Function control	122	0	4.4.10
Array manager	123	0	4.4.11
Communication port protection	124	0	4.4.12
SCHC-LPWAN setup	126	0	4.16.2.1
SCHC-LPWAN diagnostic	127	0	4.16.2.2
LoRaWAN setup	128	0	4.16.3.2.2
LoRaWAN diagnostic	129	0	4.16.3.2.5
ISO/IEC14908 Identification	130	0	4.18.2
ISO/IEC 14908 Protocol setup	131	0	4.18.3
ISO/IEC 14908 protocol status	132	1	4.18.4
ISO/IEC 14908 diagnostic	133	1	4.18.5
HS-PLC ISO/IEC 12139-1 MAC setup	140	0	4.14.2
HS-PLC ISO/IEC 12139-1 CPAS setup	141	0	4.14.3
HS-PLC ISO/IEC 12139-1 IP SSAS setup	142	0	4.14.4
HS-PLC ISO/IEC 12139-1 HDLC SSAS setup	143	0	4.14.5
LTE monitoring	151	1 0	4.7.9 5.7.10

4.3 Interface classes for parameters and measurement data

4.3.1 Data (class_id = 1, version = 0)

4.3.1.1 Overview

This IC allows modelling various data, such as configuration data and parameters. The data are identified by the attribute *logical_name*.

Data	**0…n**	**class_id = 1, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. value	CHOICE				x + 0x08
Specific methods	**m/o**				

4.3.1.2 Attribute description

4.3.1.2.1 logical_name

Identifies the “Data” object instance. See Clause 6 and IEC 62056-6-1:2023.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.1.2.2 value

Contains the data.

The data type depends on the instantiation defined by the *logical_name* and possibly from the manufacturer. It has to be chosen so, that together with the *logical_name*, an unambiguous interpretation is possible. Any simple and complex data types listed in 4.1.5 can be used, unless the choice is restricted in Clause 6.

```
CHOICE

{
 -- simple data types

null-data                   [0],
boolean                     [3],
bit-string                  [4],
double-long                 [5],
double-long-unsigned        [6],
octet-string                [9],
visible-string              [10],
utf8-string                 [12],
bcd                         [13],
integer                     [15],
long                        [16],
unsigned                    [17],
long-unsigned               [18],
long64                      [20],
long64-unsigned             [21],
enum                        [22],
float32                     [23],
float64                     [24],
date-time                   [25],
date                        [26],
time                        [27],
delta-integer               [28],
delta-long                  [29],
delta-double-long           [30],
delta-unsigned              [31],
delta-long-unsigned         [32],
delta-double-long-unsigned [33],

-- complex data types
array                       [1],
structure                   [2],
compact-array               [19]

}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.2 Register (class_id = 3, version = 0)

4.3.2.1 Overview

This IC allows modelling a process or a status value with its associated scaler and unit. “Register” objects know the nature of the process or status value. It is identified by the attribute *logical_name*.

Register		0...n	class_id = 3, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. value		CHOICE				x + 0x08
3. scaler_unit	(static)	scal_unit_type				x + 0x10
Specific methods		***m/o***				
1. reset (data)		o				x + 0x28

4.3.2.2 Attribute description

4.3.2.2.1 logical_name

Identifies the “Register” object instance. See Clause 6 and IEC 62056-6-1:2023.

4.3.2.2.2 value

Contains the current process or status value.

The data type of the value depends on the instantiation defined by *logical_name* and possibly on the choice of the manufacturer. It has to be chosen so that, together with the *logical_name*, an unambiguous interpretation of the value is possible.

When a “Register” object is used instead of a “Data” object, (with the *scaler_unit* attribute not used or with scaler = 0, unit = 255) then the data types allowed for the *value* attribute of the “Data” IC are allowed.

```
CHOICE
{
        -- simple data types

null-data               [0],
bit-string              [4],
double-long             [5],
double-long-unsigned    [6],
octet-string            [9],
visible-string          [10],
utf8-string             [12],
integer                 [15],
long                    [16],
unsigned                [17],
long-unsigned           [18],
long64                  [20],
long64-unsigned         [21],
enum                    [22],
float32                 [23],
float64                 [24],

}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.2.2.3 scaler_unit

Provides information on the unit and the scaler of the value.

```
scal_unit_type::= structure
{
scaler,
unit
}

scaler:     integer
```

This is the exponent (to the base of 10) of the multiplication factor.

REMARK If the value is not numerical, then the scaler shall be set to 0.

See Table 6 for examples for scaler_unit.

```
unit:     enum
```

See Table 5 for values of unit.

4.3.2.3 Method description

4.3.2.3.1 reset(data)

Forces a reset of the object. By invoking this method, the value is set to the default value. The default value is an instance specific constant.

```
data::= integer (0)
```

Table 5 – Enumerated values for physical units

unit::= enum	Unit	Quantity	Unit name	SI definition (comment)
(1)	a	time	year	
(2)	mo	time	month	
(3)	wk	time	week	7*24*60*60 s
(4)	d	time	day	24*60*60 s
(5)	h	time	hour	60*60 s
(6)	min	time	minute	60 s
(7)	s	time (t)	second	s
(8)	°	(phase) angle	degree	rad*180/π
(9)	°C	temperature (T)	degree-celsius	K-273.15
(10)	currency	(local) currency		
(11)	m	length (l)	metre	m
(12)	m/s	speed (v)	metre per second	m/s
(13)	m^3	volume (V) r_V, meter constant or pulse value (volume)	cubic metre	m^3
(14)	m^3	corrected volume [a]	cubic metre	m^3
(15)	m^3/h	volume flux	cubic metre per hour	$m^3/(60*60s)$
(16)	m^3/h	corrected volume flux [a]	cubic metre per hour	$m^3/(60*60s)$
(17)	m^3/d	volume flux		$m^3/(24*60*60s)$

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

unit::= enum	Unit	Quantity	Unit name	SI definition (comment)
(18)	m^3/d	corrected volume flux [a]		$m^3/(24*60*60s)$
(19)	l	volume	litre	10^{-3} m^3
(20)	kg	mass (m)	kilogram	
(21)	N	force (F)	newton	
(22)	Nm	energy	newton metre	J = Nm = Ws
(23)	Pa	pressure (p)	pascal	N/m^2
(24)	bar	pressure (p)	bar	10^5 N/m^2
(25)	J	energy	joule	J = Nm = Ws
(26)	J/h	thermal power	joule per hour	J/(60*60s)
(27)	W	active power (P)	watt	W = J/s
(28)	VA	apparent power (S)	volt-ampere	
(29)	var	reactive power (Q)	var	
(30)	Wh	active energy r_W, active energy meter constant or pulse value	watt-hour	W*(60*60s)
(31)	VAh	apparent energy r_S, apparent energy meter constant or pulse value	volt-ampere-hour	VA*(60*60s)
(32)	varh	reactive energy r_B, reactive energy meter constant or pulse value	var-hour	var*(60*60s)
(33)	A	current (I)	ampere	A
(34)	C	electrical charge (Q)	coulomb	C = As
(35)	V	voltage (U)	volt	V
(36)	V/m	electric field strength (E)	volt per metre	V/m
(37)	F	capacitance (C)	farad	C/V = As/V
(38)	Ω	resistance (R)	ohm	Ω = V/A
(39)	$\Omega m^2/m$	resistivity (ρ)		Ωm
(40)	Wb	magnetic flux (Φ)	weber	Wb = Vs
(41)	T	magnetic flux density (B)	tesla	Wb/m^2
(42)	A/m	magnetic field strength (H)	ampere per metre	A/m
(43)	H	inductance (L)	henry	H = Wb/A
(44)	Hz	frequency (f, ω)	hertz	1/s
(45)	1/(Wh)	R_W, active energy meter constant or pulse value		
(46)	1/(varh)	R_B, reactive energy meter constant or pulse value		
(47)	1/(VAh)	R_S, apparent energy meter constant or pulse value		
(48)	V^2h	volt-squared hour, r_{U2h} , volt-squared hour meter constant or pulse value	volt-squared-hours	$V^2(60*60s)$
(49)	A^2h	ampere-squared hour, r_{I2h} , ampere-squared hour meter constant or pulse value	ampere-squared-hours	$A^2(60*60s)$
(50)	kg/s	mass flux	kilogram per second	kg/s
(51)	S, mho	conductance	siemens	1/Ω
(52)	K	temperature (T)	kelvin	
(53)	$1/(V^2h)$	R_{U2h}, volt-squared hour meter constant or pulse value		
(54)	$1/(A^2h)$	R_{I2h}, ampere-squared hour meter constant or pulse value		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

unit::= enum	Unit	Quantity	Unit name	SI definition (comment)
(55)	$1/m^3$	R_V, meter constant or pulse value (volume)		
(56)		percentage	%	
(57)	Ah	ampere-hours	Ampere-hour	
…				
(60)	Wh/m^3	energy per volume	$3,6*10^3$ J/m^3	
(61)	J/m^3	calorific value, wobbe		
(62)	Mol %	molar fraction of gas composition	mole percent	(Basic gas composition unit)
(63)	g/m^3	mass density, quantity of material		(Gas analysis, accompanying elements)
(64)	Pa s	dynamic viscosity	pascal second	(Characteristic of gas stream)
(65)	J/kg	specific energy NOTE The amount of energy per unit of mass of a substance	Joule / kilogram	$m^2 . kg . s^{-2} / kg = m^2 . s^{-2}$
….				
(70)	dBm	signal strength, dB milliwatt (e.g. of GSM radio systems)		
(71)	dbµV	signal strength, dB microvolt		
(72)	dB	logarithmic unit that expresses the ratio between two values of a physical quantity		
…				
(253)		reserved		
(254)	other	other unit		
(255)	count	no unit, unitless, count		

[a] Usage of these units (14, 16 & 18) is deprecated as OBIS codes specify a corrected volume (flux). This is to avoid contradiction of units associated with selected OBIS codes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 6 – Examples for scaler_unit

Value	Scaler	Unit	Data
263788	-3	m^3	263,788 m^3
593	3	Wh	593 kWh
3467	-1	V	346,7
3467	0	V	3467 V
3467	1	V	34 670 V

4.3.3 Extended register (class_id = 4, version = 0)

4.3.3.1 Overview

This IC allows modelling a process value with its associated scaler, unit, status and capture time information. “Extended register” objects know the nature of the process value. It is described by the attribute *logical_name*.

Extended register		0...n	class_id = 4, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. value	(dyn.)	CHOICE				x + 0x08
3. scaler_unit	(static)	scal_unit_type				x + 0x10
4. status	(dyn.)	CHOICE				x + 0x18
5. capture_time	(dyn.)	octet-string				x + 0x20
Specific methods		***m/o***				
1. reset (data)		o				x + 0x38

4.3.3.2 Attribute description

4.3.3.2.1 logical_name

Identifies the “Extended register” object instance. See 6.3.1.

4.3.3.2.2 value

See the specification of the IC "Register” in 4.3.3.2.2.

4.3.3.2.3 scaler_unit

See the specification of the IC "Register" in 4.3.3.2.3.

4.3.3.2.4 status

Provides “Extended register” specific status information. The meaning of the elements of the status shall be provided for each object instance.

The data type and the encoding depend on the instantiation and possibly on the choice of the manufacturer. For the interpretation, extra information from the manufacturer may be necessary.

```
CHOICE
{
        -- simple data types

null-data               [0],
bit-string              [4],
double-long-unsigned    [6],
octet-string            [9],
visible-string          [10],
utf8-string             [12],
unsigned                [17],
long-unsigned           [18],
long64-unsigned         [21]

}
```

Def. Depending on the status type definition.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.3.2.5 capture_time

Provides an “Extended register” specific date and time information showing when the value of the attribute *value* has been captured.

octet-string, formatted as specified in 4.1.6.1 for *date-time.*

4.3.3.3 Method description

4.3.3.3.1 reset (data)

This method forces a reset of the object. By invoking this method, the attribute *value* is set to the default value. The default value is an instance specific constant.

The attribute *capture_time* is set to the time of the reset execution.

data::= integer (0)

4.3.4 Demand register (class_id = 5, version = 0)

4.3.4.1 Overview

This IC allows modelling a demand value with its associated scaler, unit, status and time information. A “Demand register” object measures and computes a *current_average_value* periodically and it stores a *last_average_value.* The time interval *T* over which the demand is calculated is defined by specifying *number_of_periods* and *period*. See Figure 8.

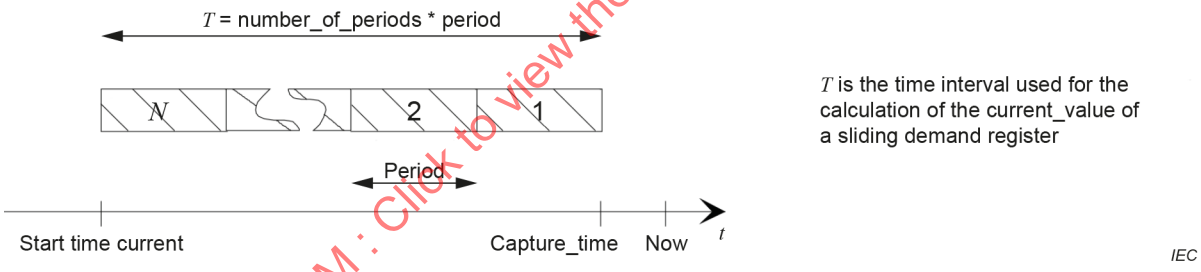


Figure 8 – The time attributes when measuring sliding demand

The demand register delivers two types of demand: *current_average_value* and *last_average_value* (see Figure 9 and Figure 10).

“Demand register” objects know the nature the of process value, which is described by the attribute *logical_name*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

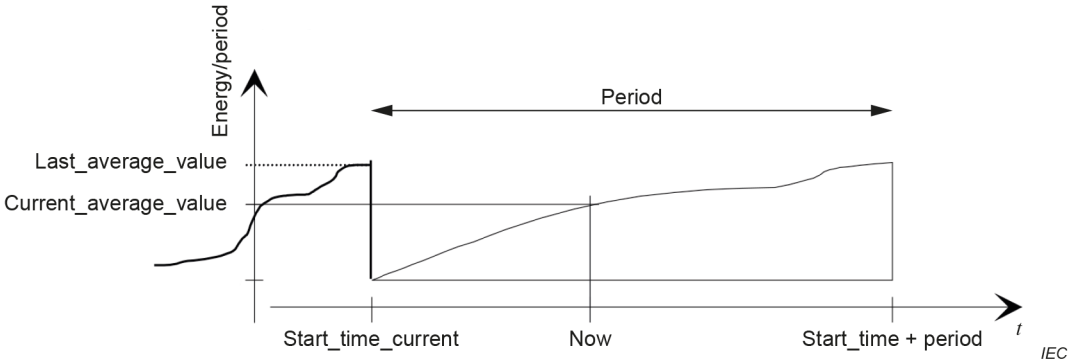


Figure 9 – The attributes in the case of block demand

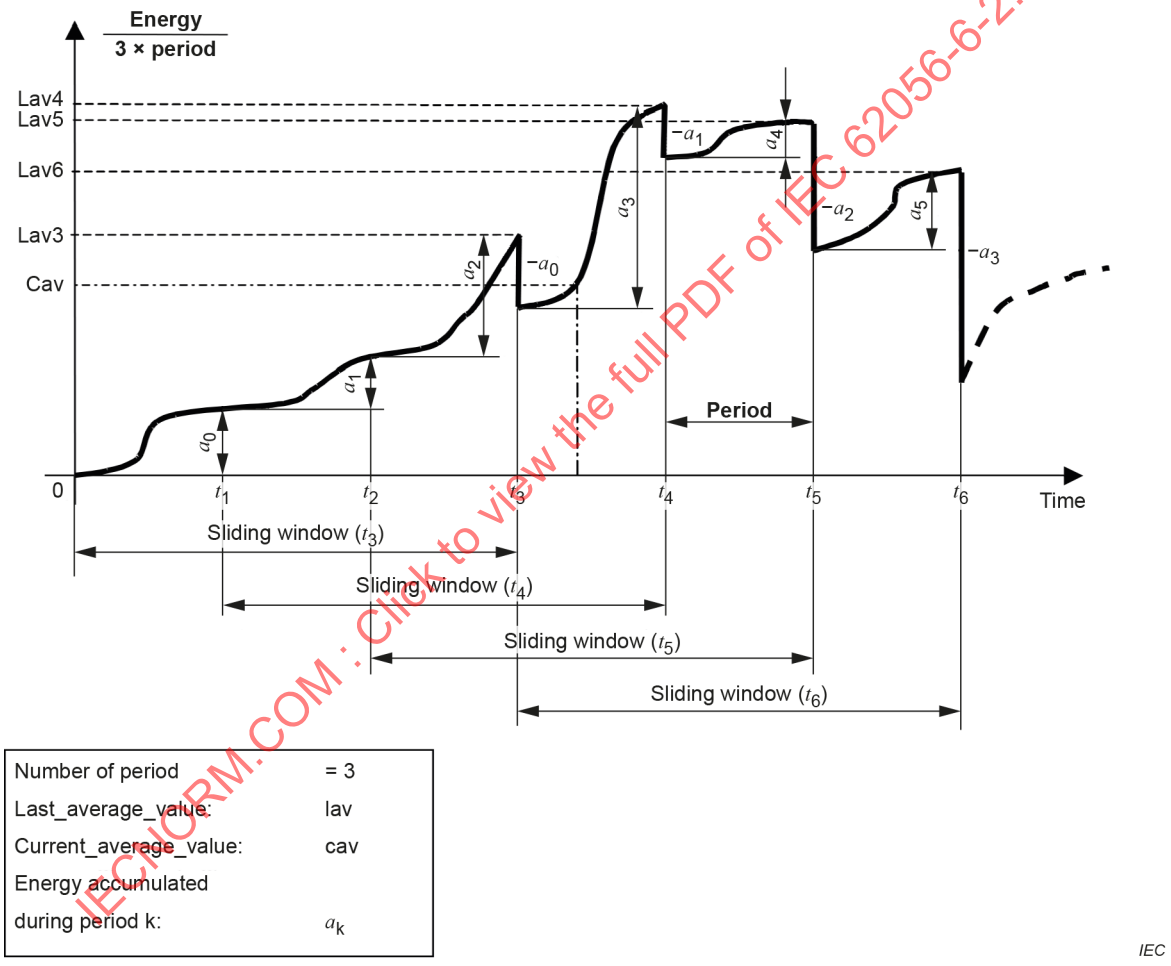


Figure 10 – The attributes in the case of sliding demand (number of periods = 3)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Demand register		0...n	class_id = 5, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. current_average_value	(dyn.)	CHOICE			0	x + 0x08
3. last_average_value	(dyn.)	CHOICE			0	x + 0x10
4. scaler_unit	(static)	scal_unit_type				x + 0x18
5. status	(dyn.)	CHOICE				x + 0x20
6. capture_time	(dyn.)	octet-string				x + 0x28
7. start_time_current	(dyn.)	octet-string				x + 0x30
8. period	(static)	double-long-unsigned	1			x + 0x38
9. number_of_periods	(static)	long-unsigned	1		1	x + 0x40
Specific methods		***m/o***				
1. reset (data)		o				x + 0x48
2. next_period (data)		o				x + 0x50

4.3.4.2 Attribute description

4.3.4.2.1 logical_name

Identifies the “Demand register” object instance. See 6.3.1 and IEC 62056-6-1:2023.

4.3.4.2.2 current_average_value

Provides the current value (running demand) of the energy accumulated since *start_time*, divided by *number_of_periods*period*.

The data type of the value depends on the instantiation defined by *logical_name* and possibly on the choice of the manufacturer. The type has to be chosen so that – together with the *logical_name* – unambiguous interpretation of the value is possible.

If a quantity other than energy is measured, other calculation methods may apply (for example for calculating average values of voltage or current).

CHOICE For data types, see “Register” value attribute (4.3.2.2.2).

4.3.4.2.3 last_average_value

Provides the value of the energy accumulated (over the last number_of_periods*period) divided by number_of_periods*period. The energy of the current (not terminated) period is not considered by the calculation.

If a quantity other than energy is measured, other calculation methods may apply (for example for calculating average values of voltage or current).

CHOICE For data types, see “Register” value attribute (4.3.2.2.2).

4.3.4.2.4 scaler_unit

See the specification of IC “Register” in 4.3.2.2.3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.4.2.5 status

Provides “Demand register” specific status information. The data type and the encoding depend on the instantiation and possibly on the choice of the manufacturer. For the interpretation, extra information from the manufacturer may be necessary.

CHOICE For data types, see “Extended register” IC status attribute (4.3.3.2.4).

Def. Depending on the status type definition.

4.3.4.2.6 capture_time

Provides the date and time when the last_average_value has been calculated.

octet-string, formatted as specified in 4.1.6.1 for date-time.

4.3.4.2.7 start_time_current_

Provides the date and time when the measurement of the current_average_value has been started.

octet-string, formatted as specified in 4.1.6.1 for date-time.

4.3.4.2.8 period

Period is the interval between two successive updates of the last_average_value. (number_of_periods*period is the denominator for the calculation of the demand).

double-long-unsigned Measuring period in seconds

The behaviour of the meter after writing a new value to this attribute shall be specified by the manufacturer.

4.3.4.2.9 number_of_periods

The number of periods used to calculate the last_average_value. number_of_periods >= 1

– number_of_periods > 1 indicates that the last_average_value represents “sliding demand”,
– number_of_periods = 1 indicates that the last_average_value represents "block demand".

The behaviour of the meter after writing a new value to this attribute shall be specified by the manufacturer.

4.3.4.3 Method description

4.3.4.3.1 reset (data)

This method forces a reset of the object. Activating this method provokes the following actions:

– the current period is terminated;
– the current_average_value and the last_average_value are set to their default values;
– the capture_time and the start_time_current are set to the time of the execution of reset (data).

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.4.3.2 next_period (data)

This method is used to trigger the regular termination (and restart) of a period. Closes (terminates) the current measuring period. Updates capture_time and start_time and copies current_average_value to last_average_value, sets current_average_value to its default value. Starts the next measuring period.

REMARK The old last_average_value (and capture_time) can be read during the time "period". The old current_average_value is not available any more at the interface.

data::= integer (0)

4.3.5 Register activation (class_id = 6, version = 0)

4.3.5.1 Overview

This IC allows modelling the handling of different tariffication structures. To each "Register activation" object, groups of "Register", "Extended register" or "Demand register" objects, modelling different kind of quantities (for example active energy, active demand, reactive energy, etc.) are assigned. Subgroups of these registers, defined by the *activation masks* define different tariff structures (for example day tariff, night tariff). One of these activation masks, the *active_mask*, defines which subset of the registers, assigned to the "Register activation" object instance is active. Registers not included in the *register_assignment* attribute of any "Register activation" object are always enabled by default.

Register activation		**0...n**	**class_id = 6, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. register_assignment	(static)	array				x + 0x08
3. mask_list	(static)	array				x + 0x10
4. active_mask	(dyn.)	octet-string				x+ 0x18
Specific methods		***m/o***				
1. add_register (data)		o				x + 0x30
2. add_mask (data)		o				x + 0x38
3. delete_mask (data)		o				x + 0x40

4.3.5.2 Attribute description

4.3.5.2.1 logical_name

Identifies the "Register activation" object instance. See 6.2.11.

4.3.5.2.2 register_assignment

Specifies an ordered list of COSEM objects assigned to the "Register activation" object. The list may contain different kinds of COSEM objects, for example instances of "Register", "Extended register" or "Demand register".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

array object_definition

object_definition::= structure

{
class_id: long-unsigned,
logical_name: octet-string
}

4.3.5.2.3 mask_list

Specifies a list of register activation masks. Each entry (mask) is identified by its mask_name and contains an array of indices referring to the registers assigned to the mask (the first object in register_assignment is referenced by index 1, the second object by index 2, ...).

array register_act_mask

register_act_mask::= structure

{
mask_name: octet-string,
index_list: index_array
}

mask_name has to be uniquely defined within the object
index_array::= array unsigned

4.3.5.2.4 active_mask

Defines the currently active mask. The mask is defined by its mask_name (see mask_list, 4.3.5.2.3).

The active_mask defines the registers currently enabled; all other registers listed in the register_assignment are disabled.

4.3.5.3 Method description

4.3.5.3.1 add_register (data)

Adds one more register to the attribute register_assignment. The new register is added at the end of the array; i.e. the newly added register has the highest index. The indices of the existing registers are not modified.

data::= structure
{
class_id: long-unsigned,
logical_name: octet-string
}

4.3.5.3.2 add_mask (data)

Adds another mask to the attribute mask_list. If there exists already a mask with the same name, the existing mask will be overwritten by the new mask.

data::= register_act_mask (see 4.3.5.3.1)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.5.3.3 delete_mask (data)

Deletes a mask from the attribute mask_list. The mask is defined by its mask name.

data::= octet-string (mask_name)

4.3.6 Profile generic (class_id = 7, version = 1)

4.3.6.1 Overview

This IC provides a generalized concept allowing to store, sort and access data groups or data series, called *capture objects*. Capture objects are appropriate attributes or elements of (an) attribute(s) of COSEM objects. The capture objects are collected periodically or occasionally.

A profile has a *buffer* to store the captured data. To retrieve only a part of the buffer, either a value range or an entry range may be specified, asking to retrieve all entries that fall within the range specified.

The list of *capture objects* defines the values to be stored in the *buffer* (using auto capture or the method *capture*). The list is defined statically to ensure homogenous buffer entries (all entries have the same size and structure). If the list of capture objects is modified, the *buffer* is cleared. If the buffer is captured by other “Profile generic” objects, their *buffer* is cleared as well, to guarantee the homogeneity of their *buffer* entries.

The *buffer* may be defined as sorted by one of the *capture objects*, e.g. the clock, or the entries are stacked in a “last in first out” order. For example, it is very easy to build a “maximum demand register” with a one entry deep sorted profile capturing and sorted by a “Demand register” *last_average_value* attribute. It is just as simple to define a profile retaining the three largest values of some period.

The size of profile data is determined by three parameters:

a) the number of entries filled (*entries_in_use*). This will be zero after clearing the profile;

b) the maximum number of entries to retain (*profile_entries*). If all entries are filled and a capture () request occurs, the least important entry (according to the requested sorting method) will get lost. This maximum number of entries may be specified. Upon changing it, the buffer will be adjusted;

c) the physical limit for the buffer. This limit typically depends on the objects to capture. The object will reject an attempt of setting the maximum number of entries that is larger than physically possible.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Profile generic		0...n	class_id = 7, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. buffer	(dyn.)	compact-array or array				x + 0x08
3. capture_objects	(static)	array				x + 0x10
4. capture_period	(static)	double-long-unsigned				x + 0x18
5. sort_method	(static)	enum			1	x + 0x20
6. sort_object	(static)	capture_object_definition				x + 0x28
7. entries_in_use	(dyn.)	double-long-unsigned	0		0	x + 0x30
8. profile_entries	(static)	double-long-unsigned	1		1	x + 0x38
Specific methods		***m/o***				
1. reset (data)		o				x + 0x58
2. capture (data)		o				x + 0x60
3. *reserved from previous versions*		o				
4. *reserved from previous versions*		o				

4.3.6.2 Attribute description

4.3.6.2.1 logical_name

Identifies the “Profile generic” object instance. For examples, see 6.2.17, 6.2.19, 6.2.21, 6.2.22, 6.2.43, 6.2.46, 6.2.49, 6.2.50, 6.2.61, 6.2.62, 6.2.64, 6.2.65, 6.2.66, 6.2.67, 6.3.2, 6.3.6, etc.

4.3.6.2.2 buffer

Contains a sequence of entries. Each entry contains values of the captured objects.

```
compact-array or array        entry

entry:= structure

{
        CHOICE
        {
                -- simple data types
                null-data                [0],
                boolean                  [3],
                bit-string               [4],
                double-long              [5],
                double-long-unsigned     [6],
                octet-string             [9],
                visible-string           [10],
                utf8-string              [12],
                bcd                      [13],
                integer                  [15],
                long                     [16],
                unsigned                 [17],
                long-unsigned            [18],
                long64                   [20],
                long64-unsigned          [21],
                enum                     [22],
                float32                  [23],
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
                    float64                     [24],
                    date-time                   [25],
                    date                        [26],
                    time                        [27],
                    delta-integer               [28],
                    delta-long                  [29],
                    delta-double-long           [30],
                    delta-unsigned              [31],
                    delta-long-unsigned         [32],
                    delta-double-long-unsigned  [33],

                    -- complex data types
                    array                       [1],
                    structure                   [2],
                    compact-array               [19]
            }
    }
```

The number and the order of the elements of the structure holding the entries is the same as in the definition of the *capture_objects*. The *buffer* is filled by auto captures or by subsequent calls of the method *capture*. The sequence of the entries within the array is ordered according to the *sort_method* specified.

Default: The *buffer* is empty after reset.

REMARK 1 Reading the entire *buffer* delivers only those entries, which are “in use”.

REMARK 2 The value of a captured object may be replaced by “null-data” if it can be unambiguously recovered from the previous value (for example for time: if it can be calculated from the previous value and *capture_period*; or for a value: if it is equal to the previous value).

selective access (see 4.1.4.17) to the attribute buffer may be available (optional). The selective access parameters are as defined in 4.3.6.2.9.

4.3.6.2.3 capture_objects

Specifies the list of capture objects that are assigned to the object instance.

Upon a call of the *capture* (data) method or automatically in defined intervals, the selected attributes are copied into the *buffer* of the profile.

```
array       capture_object_definition

capture_object_definition::= structure
{
        class_id:               long-unsigned,
        logical_name:           octet-string,
        attribute_index:        integer,
        data_index:             long-unsigned
}
```

– where attribute_index is a pointer to the attribute within the object identified by its class_id and *logical_name*. Attribute_index 1 refers to the 1st attribute (i.e. the *logical_name*), attribute_index 2 to the 2nd, etc.); attribute_index 0 refers to all public attributes;

– where data_index is a pointer selecting a specific element of the attribute. The first element in the attribute structure is identified by data_index 1. If the attribute is not a structure, then the data_index has no meaning. If the capture object is the buffer of a profile, then the data_index identifies the captured object of the buffer (i.e. the column) of the inner profile;

data_index 0: references the whole attribute.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.6.2.4 capture_period

>= 1: Automatic capturing assumed. Specifies the capturing period in seconds.

0: No automatic capturing; capturing is triggered externally or capture events occur asynchronously.

4.3.6.2.5 sort_method

If the profile is unsorted, it works as a “first in first out” buffer (it is hence sorted by capturing, and not necessarily by the time maintained in the “Clock” object). If the *buffer* is full, the next call to *capture* () will push out the first (oldest) entry of the *buffer* to make space for the new entry.

If the profile is sorted, a call to *capture* () will store the new entry at the appropriate position in the buffer, moving all following entries and probably losing the least interesting entry. If the new entry would enter the *buffer* after the last entry and if the *buffer* is already full, the new entry will not be retained at all.

enum:
(1) fifo (first in first out),
(2) lifo (last in first out),
(3) largest,
(4) smallest,
(5) nearest_to_zero,
(6) farest_from_zero

Def. fifo

4.3.6.2.6 sort_object

If the profile is sorted, this attribute specifies the register or clock that the ordering is based upon.

capture_object_definition See 4.3.6.2.3

Def. no object to sort by (only possible with sort_method fifo or lifo)

NOTE If the sort_method is FIFO or LIFO, then all elements of the capture_object_definition specifying the sort object can be zero.

4.3.6.2.7 entries_in_use

Counts the number of entries stored in the buffer. After a call of the *reset* () method, the buffer does not contain any entries, and this value is zero. Upon each subsequent call of the *capture* () method, this value will be incremented up to the maximum number of entries that will get stored (see *profile_entries,* 4.3.6.2.8).

double-long-unsigned 0…profile_entries

Def. 0

4.3.6.2.8 profile_entries

Specifies how many entries shall be retained in the buffer.

double-long-unsigned 1…(limited by physical size)

Def. 1

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.6.2.9 Parameters for selective access to the buffer attribute

Table 7 defines the parameters required for selective access to the buffer.

Table 7 – Parameters for selective access to the buffer attribute

Access selector	Access parameter	Comment
1	range_descriptor	Only buffer elements corresponding to the range_descriptor shall be returned in the response.
2	entry_descriptor	Only buffer elements corresponding to the entry_descriptor shall be returned in the response.

range_descriptor::= structure
{ restricting_object: capture_object_definition

(Defines the capture_object restricting the range of entries to be retrieved. Only simple data types are allowed.)

from_value: (Oldest or smallest entry to retrieve)

CHOICE

{ -- simple data types
double-long [5],
double-long-unsigned [6],
octet-string [9],
visible-string [10],
utf8-string [12],
integer [15],
long [16],
unsigned [17],
long-unsigned [18],
long64 [20],
long64-unsigned [21],
float32 [23],
float64 [24],
date-time [25],
date [26],
time [27]
}

to_value: (Newest or largest entry to retrieve)

CHOICE
{as for 'from_value' above}

selected_values: array capture_object_definition

(List of columns to retrieve. If the array is empty (has no entries), all captured data are returned. Otherwise, only the columns specified in the array are returned. The type *capture_object_definition* is specified above (*capture_objects*)).
}

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
entry_descriptor::= structure
{
    from_entry:          double-long-unsigned   first entry to retrieve,
    to_entry:            double-long-unsigned   last entry to retrieve
    (if to_entry == 0: highest possible entry),

    from_selected_value: long-unsigned  index of first value to retrieve,
    to_selected_value:   long-unsigned  index of last value to retrieve

    (if to_selected_value == 0: highest possible selected_value)
}
```

NOTE 1 from_entry and to_entry identify the lines, from_selected_value to_selected_value identify the columns of the buffer to be retrieved.

NOTE 2 Numbering of entries and selected values starts from 1.

4.3.6.3 Method description

4.3.6.3.1 reset (data)

Clears the *buffer*. It has no valid entries afterwards; *entries_in_use* is zero after this call. This call does not trigger any additional operations on the capture objects. Specifically, it does not reset any attributes captured.

```
data::= integer (0)
```

4.3.6.3.2 capture(data)

Copies the values of the objects to capture into the *buffer* by reading each capture object. Depending on the *sort_method* and the actual state of the *buffer* this produces a new entry or a replacement for the less significant entry. As long as not all entries are already used, the *entries_in_use* attribute will be incremented.

This call does not trigger any additional operations within the capture objects such as *capture* () or *reset* ().

Note, that if more than one attribute of an object need to be captured, they have to be defined one by one on the list of *capture_objects*. If the attribute_index = 0, all attributes are captured.

```
data::= integer (0)
```

4.3.6.4 Behaviour of the object after modification of certain attributes

Any modification of one of the *capture_objects* describing the static structure of the *buffer* will automatically call a *reset* () and this call will propagate to all other profiles capturing this profile.

If writing to *profile_entries* is attempted with a value too large for the buffer, it will be rejected.

4.3.6.5 Restrictions

When defining the *capture objects*, circular reference to the profile shall be avoided.

4.3.6.6 Profile used to define a subset of preferred readout values

By setting *profile_entries* to 1, a “Profile generic” object can be used to define a set of preferred readout values. See also 6.2.19. Setting *capture_period* to 1 ensures that the values are updated every second.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.7 Utility tables (class_id = 26, version = 0)

4.3.7.1 Overview

This IC allows encapsulating ANSI C12.19 table data. Each “table” is represented by an instance of this IC, identified by its *logical name*.

Utility tables		**0...n**	**class_id = 26, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. table_ID	(static)	long-unsigned				x + 0x08
3. length		double-long-unsigned				x + 0x10
4. buffer		octet-string				x + 0x18
Specific methods		***m/o***				

4.3.7.2 Attribute description

4.3.7.3 logical_name

Identifies the “Utility tables” object instance. See 6.2.40.

4.3.7.4 table_ID

Table number. This table number is as specified in the ANSI standard and may be either a standard table or a manufacturer’s table.

4.3.7.5 length

Number of octets in table buffer.

4.3.7.6 buffer

Contents of the table.

Selective access (see 4.1.4.17) to the attribute *buffer* may be available (optional). The selective access parameters are defined in Table 8.

Table 8 – Parameters for selective access to the buffer attribute

Access selector	Parameter	Comment
1	offset_access	Access to table by offset and count using offset_selector for parameter data.
2	index_access	Access to table by element id and number of elements using index_selector for parameter data.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

offset_selector::= structure

{

Offset:	double-long-unsigned	Offset in octets to the start of access area, relative to the start of the table.
Count:	long-unsigned	Number of octets requested or transferred

}

index_selector::= structure

{

Index:	array long-unsigned	Sequence of indices to identify elements within the table's hierarchy.
Count:	long-unsigned	Number of elements requested or transferred. Values of count greater than 1 return up to that many elements. A value of zero, when given in the context of a request, refers to the entire sub-tree of the hierarchy starting at the selection point.

}

4.3.8 Register table (class_id = 61, version = 0)

4.3.8.1 Overview

This IC allows to group homogenous entries, identical attributes of multiple objects, which are all instances of the same IC, and in their *logical_name* (OBIS code) the value in value groups A to D and F is identical. The possible values in value group E are defined in IEC 62056-6-1:2023 in a tabular form: the table header defines the common part of the OBIS code and each table cell defines one possible value of value group E. A "Register table" object may capture attributes of some or all of those objects.

NOTE 1 Some examples are the "Extended phase angle measurement" table, see Table 17 or the "UNIPEDE voltage dip quantities" table, see Table 19.

NOTE 2 If more complex functionality is needed, the "Profile generic" IC can be used.

Register table		**0...n**	**class_id = 61, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. table_cell_values	(dyn.)	compact-array or array				x + 0x08
3. table_cell_definition	(static)	structure				x + 0x10
4. scaler_unit	(static)	scaler_unit_type				x + 0x18
Specific methods		***m/o***				
1. reset (data)		o				x + 0x28
2. capture (data)		o				x + 0x30

4.3.8.2 Attribute description

4.3.8.2.1 logical_name

Identifies the "Register table" object instance.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

When the format of the *logical_name* is A.B.C.D.255.F; the values A to D and F define the common part of the logical name of the objects, the attributes of which are captured. Only one attribute of the objects concerned can be captured (for example the *value* attribute).

When the format of the *logical_name* is A.B.98.10.x.255, several instances of the "Register table" IC can be used to capture different attributes of the objects concerned. The value group E numbers the instances. See IEC 62056-6-1:2023, 6.4.

4.3.8.2.2 table_cell_values

Holds the value of the attributes captured, as they would be returned to a GET or Read .request to the individual attributes.

```
compact array or array          table_cell_entry

table_cell_entry::=    CHOICE
{
        -- simple data types
        null-data                   [0],
        bit-string                  [4],
        double-long                 [5],
        double-long-unsigned        [6],
        octet-string                [9],
        visible-string              [10],
        utf8-string                 [12],
        bcd                         [13],
        integer                     [15],
        long                        [16],
        unsigned                    [17],
        long-unsigned               [18],
        long64                      [20],
        long64-unsigned             [21],
        float32                     [23],
        float64                     [24],

        -- complex data types
        structure                   [2]
}
```

If the captured attribute is attribute_0, redundant values may be replaced by "null-data", if their value can be unambiguously recovered (for example *scaler_unit*).

4.3.8.2.3 table_cell_definition

Specifies the list of attributes captured in the register table.

structure

```
{
        class_id:           long-unsigned,
        logical_name:       octet-string,
        group_E_values:     array unsigned,
        attribute_index:    integer
}
```

where:

– class_id defines the common class_id of the objects the attributes of which are captured;

– logical_name contains the common logical name of the objects, with E = 255 (wildcard);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– group_E_values contain the list of cell identifiers, of type *unsigned*, as defined in the respective table of IEC 62056-6-1:2023;
– attribute_index is a pointer to the attribute within the object. attribute_index 0 refers to all public attributes.

If the *logical_name* of the "Register table" object is in the format A.B.C.D.255.F and the defined attribute of all objects identified in the respective table in IEC 62056-6-1:2023 are captured, then attribute 3 may not be accessible. In this case:

– the class_id shall be 1 "Data", 3 "Register" or 4 "Extended register";
– the *logical_name* of the objects to be captured is defined by the *logical_name* of the "Register table" object and the respective table inIEC 62056-6-1:2023.

the attribute index shall be 2 (value).

4.3.8.2.4 scaler_unit

See the description of IC "Register", 4.3.2.2.3.

In the case when "value" attributes of "Register" or "Extended register" objects are captured, the *scaler_unit* shall be common for all objects and this attribute shall hold a copy.

If other attributes or ICs are captured, the *scaler_unit* attribute has no meaning and shall be inaccessible.

4.3.8.3 Method description

4.3.8.3.1 reset (data)

Clears the *table_cell_values*. It has no effect on the attributes captured.

data::= integer (0)

4.3.8.3.2 capture (data)

Copies the values of the attributes into the *table_cell_values*. If the attribute_index = 0, all attributes are captured.

4.3.8.4 Behaviour of the object after modification of the table_cell_definition attribute

Any modification to this attribute will automatically call the *reset* (data) method and this will propagate to all profiles capturing this object.

If writing to *table_cell_definition* is attempted with a value too large the buffer holding the *table_cell_values* attribute, it will be rejected.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.9 Status mapping (class_id = 63, version = 0)

4.3.9.1 Overview

This IC allows modelling the mapping of bits in a status word to entries in a reference table.

Status mapping		**0...n**	**class_id = 63, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. status_word	(dyn.)	CHOICE				x + 0x08
3. mapping_table	(static)	structure				x + 0x10
Specific methods		***m/o***				

4.3.9.2 Attribute description

4.3.9.2.1 logical_name

Identifies the “Status mapping” object instances. See 6.2.46, 6.2.49, 6.2.50, 6.2.55 and 6.3.7.

4.3.9.2.2 status_word

Contains the current value of the status word.

```
CHOICE
{
bit-string                [4],
double-long-unsigned      [6],
octet-string              [9],
visible-string,           [10],
utf8-string               [12],
unsigned                  [17],
long-unsigned             [18],
long64-unsigned           [21]
}
```

The size of the *status_word* is n*8 bits, the maximum size is 65 536 bits.

Manufacturers may choose any of the types listed above. However, the status word is always interpreted as a bit-string.

4.3.9.2.3 mapping_table

Contains the mapping of the status_word to the positions in the reference table.

```
structure
{
ref_table_id:             unsigned,
ref_table_mapping:        CHOICE
      {
            long-unsigned             [18],
            array long-unsigned       [1]
      }
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

– ref_table_id identifies the reference status table;

– if the “long-unsigned” choice is taken, the value points to an entry in the reference table. This entry is mapped to the leading bit of the status word. The next entry is mapped to the next bit and so on. The last entry that is mapped to the trailing bit is determined by the length of the status word;

– if the “array” choice is taken, the elements of the array point to entries in the reference status table. The order of the elements in the array corresponds to the position in the status word. The first element in the array maps the table entry referenced to the leading bit and the last element to the trailing bit.

4.3.10 Compact data (class_id: 62, version: 1)

4.3.10.1 Overview

NOTE This version 1 supports both relative and absolute selective access.

Instances of the “Compact data” IC allow capturing the values of COSEM object attributes as determined by the *capture_objects* attribute. Capturing can take place:

- on an external trigger (explicit capturing); or
- upon reading the compact_buffer attribute (implicit capturing)

as determined by the *capture_method* attribute.

The values are stored in the *compact_buffer* attribute as an octet-string.

The set of data types is identified by the *template_id* attribute. The data type of each attribute captured is held by the *template_description* attribute.

The client can reconstruct the data in the uncompacted form – i.e. including the COSEM attribute descriptor, the data type and the data values – using the *capture_objects*, *template_id* and *template_description* attributes.

Compact data		**0...n**	**class_id = 62, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. compact_buffer	(dyn.)	octet-string				x + 0x08
3. capture_objects	(static)	array				x + 0x10
4. template_id	(static)	unsigned				x + 0x18
5. template_description	(dyn.)	octet-string				x + 0x20
6. capture_method	(static)	enum				x + 0x28
Specific methods		***m/o***				
1. reset (data)		o				x + 0x38
2. capture (data)		o				x + 0x40

4.3.10.2 Attribute description

4.3.10.2.1 logical_name

Identifies the “Compact data” object instance. See 6.2.41.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.10.2.2 compact_buffer

Contains the values of the attributes captured as an *octet-string*.

When the data captured is of type *octet-string, bit-string, visible-string, utf8-string* or *array* the length is also included here.

4.3.10.2.3 capture_objects

Specifies the list of COSEM object attributes that are assigned to the “Compact data” object instance.

When defining the *capture_object* attribute, circular references shall be avoided.

The *template_id* attribute shall be the first element in the *capture_objects* array.

Upon an explicit or implicit invocation of the *capture* (data) method the values of the selected attributes are captured into the *compact_buffer*.

Two, mutually exclusive selective access mechanisms are available:

– relative selective access, i.e. entries defined relative to current date or entry are returned: this mechanism is controlled by the *data_index* element; or

– absolute selective access, i.e. entries in an explicitly defined date range or entry range are returned: this mechanism is controlled by the *restriction* element.

```
array capture_object_definition

capture_object_definition::=   structure
{
        class_id:              long-unsigned,
        logical_name:          octet-string,
        attribute_index:       integer,
        data_index:            long-unsigned,
        restriction:           restriction_element
}
```

Where:

– attribute_index is a pointer to the attribute within the object, identified by class_id and logical_name; attribute_index 1 refers to the 1st attribute (i.e. the *logical_name*), attribute_index 2 to the 2nd attribute etc.; attribute_index 0 refers to all public attributes;

– data_index is a pointer selecting one or several specific elements of an attribute with a complex data type (structure or array):

- if the data type of the attribute is simple, then data_index has no meaning;
- if the data type of the attribute is a structure or an array, then data_index points to one or several specific elements in the structure or array;
- when the attribute is the *buffer* of a “Profile generic” object, the data_index carries selective access parameters relative to current date or entry.

data_index:	**MS-Byte**		**LS-Byte**
	Upper nibble	Lower nibble	

- 0x0000 = identifies the whole attribute;
- 0x0001 to 0x0FFF = identifies one element in the complex attribute. The first element in the complex attribute is identified by data_index 1;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- 0x1000 to 0xFFFF = selective access to the array holding the *buffer* of a “Profile generic” object. The data-index selects entries within a number of last (recent) time periods, or a number of last (recent) entries, as well as the columns in the array.

The encoding is specified in Table 21.

When the attribute is the buffer of a “Profile generic” object, then restriction_element specifies selective access parameters in an explicitly defined date range or entry range.

```
restriction_element::=   structure
{
        restriction_type:   enum:
                            (0) none,
                            (1) restriction by date,
                            (2) restriction by entry
        restriction_value: CHOICE
        {
                null-data,                  // no restrictions apply
                restriction_by_date,
                restriction_by_entry
        }
}

restriction_by_date::=structure
{
        from_date:          octet-string,
        to_date:            octet-string
}

restriction_by_entry::=structure
{
        from_entry:         double-long-unsigned,
        to_entry:           double-long-unsigned
}
```

– restriction_element defines absolute selective access to a “Profile generic” *buffer* by date range (from_date to to_date) or by entries (from_entry to to_entry). To use this absolute selective access mechanism, data_index shall be set to 0x0000;

restriction_element is composed of restriction_type and restriction_value:

- for restriction by date range the restriction_type element holds (1) restriction by date and the restriction_value element holds restriction_by_date structure;
- for restriction by entries the restriction_type element holds (2) restriction by entry and the restriction_value element holds restriction_by_entry structure;

otherwise, the restriction_type element holds (0) none and the restriction_value element holds null-data. This choice shall be taken also if relative selective access is to be used.

4.3.10.2.4 template_id

Contains the identifier of the template.

It shall uniquely identify the instance of the “Compact data” IC and the *template_description.*

4.3.10.2.5 template_description

Provides the data type of each attribute captured. It is an *octet-string* generated automatically by the server upon the programming of the *capture_objects* and it has the following structure:

– the first octet is 0x02 (the tag of a structure);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– this is followed by the number of elements in the structure – the same as the number of elements in the *capture_objects* array – encoded as a variable length integer;

– this is followed by the data type of each attribute, in the same order as in the *capture_object* array:

  - in the case of attributes with simple data type, the data type is represented by a single octet, carrying the tag of the data type. In the case of *bit-string* [4], *octet-string* [9], *visible-string* [10], *utf8-string* [12] the length of the string is part of the data held in the *compact_buffer*;
  - in the case of an *array* [1], the data type is represented by a single octet 0x01. This is followed by the type description of the elements in the array. The number of elements in the array is part of the data held in the *compact_buffer*;

in the case of a *structure* [2], the data type is represented by a single octet 0x02, followed by the number of elements inside the structure followed by the tag of each element of the structure.

4.3.10.2.6 capture_method

Defines the way the *compact_buffer* is updated.

enum

(0) Capture upon invoking the *capture* (data) method. This may occur remotely or locally (explicit capturing),

(1) Capture upon reading the *compact_buffer* attribute (implicit capturing).

4.3.10.3 Method description

4.3.10.3.1 reset (data)

Clears the *compact_buffer*. After invoking this method the buffer holds an octet-string of 0 length until a new capture takes place.

This call does not trigger any additional operations of the capture objects. Specifically, it does not reset any captured attributes.

data::= integer(0)

4.3.10.3.2 capture (data)

Copies the values of the attributes into the *compact_buffer* by reading each capture object.

This call does not trigger any additional operations within the capture objects such as capture () or reset ().

data::= integer(0)

4.3.10.4 Behaviour of the object after modification of certain attributes

Any modification of the *capture_objects* shall reset the *compact_buffer* and automatically update the *template_description*.

4.3.10.5 Restrictions

When defining the *capture_object* attribute, circular references shall be avoided.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.10.6 Examples for using compact data

4.3.10.6.1 Daily billing data

Table 9 shows the daily billing data that are captured – together with the mandatory *template_id* – to the *compact_buffer* attribute of a “Compact data” object.

Table 9 – Daily billing data

Data	class _id	Logical name	attribute_ id	data_ index	Size (bytes)	Type	Value
1	2	3	4	5	6	7	8
Template Id	62	0-0:66.0.0.255	4	0	1	unsigned	0
Unix time	1	0-0:1.1.0.255	2	0	4	double-long-unsigned	1374573317
Operating status	1	0-0:96.5.0.255	2	0	1	unsigned	0x29
Error register	1	0-0:97.97.0.255	2	0	1	unsigned	0x18
Total index	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	6422483
Index F1	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	865234
Index F2	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	1234567
Index F3	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	2345678
Activity calendar name	20	0-0:13.0.0.255	2	0	6	octet-string	“ABCDEF”
Event counter	1	0-0:96.15.1.255	2	0	2	long-unsigned	7890

Table 10 shows the attributes of the “Compact data“ object.

Table 10 – Attributes of the “Compact data“ object

capture_objects (array)	For the elements of the array, see columns 2, 3, 4 and 5 of Table 9.
template_id (unsigned)	0
template_description (octet-string)	-- For the data types, see column 7 of Table 9. 02 0A 11 06 11 11 06 06 06 06 09 12
compact_buffer (octet-string) 32 bytes)	-- For the values see column 8 of Table 9. 00 51EE5305 29 18 0061FFD3 000D33D2 0012D687 0023CACE 06414243444546 1ED2

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

For comparison, the A-XDR encoding of the same data as if they were accessed using a GET-WITH-LIST service is shown in Table 11. Only the encoding of the result (SEQUENCE OF Get-Data-Result) is shown.

Table 11 – A-XDR encoding of the data (SEQUENCE OF Get-Data-Result)

Encoding	Explanation	Length
09	**SEQUENCE of 9 elements**	1
00 06 51EE5305	double-long-unsigned	6
00 11 29	unsigned	3
00 11 18	unsigned	3
00 06 0061FFD3	double-long-unsigned	6
00 06 000D33D2	double-long-unsigned	6
00 06 0012D687	double-long-unsigned	6
00 06 0023CACE	double-long-unsigned	6
00 09 06 414243444546	octet-string of length 6	9
00 12 1ED2	long-unsigned	4
	Total	50 bytes
NOTE The leading 00-s in each element are there to indicate the CHOICE "Data" in Get-Data-Result.		

4.3.10.6.2 Diagnostic and Alarm data

Table 12 shows the diagnostic and alarm data that are captured – together with the mandatory *template_id* – to the *compact_buffer* attribute of a "Compact data" object.

Table 12 – Diagnostic and Alarm data

Data	class _id	Logical name	attribute id	data_ index	Size (bytes)	Type	Value
1	2	3	4	5	6	7	8
Template Id	62	0-0:66.0.0.255	4	0	1	unsigned	1
Current Diagnostic	3	7-0:96.5.1.255	2	0	2	long-unsigned	0x4200
Daily Diagnostic	3	7-1:96.5.1.255	2	0	2	long-unsigned	0x4108
Billing Period Diagnostic	1	7-2:96.5.1.255	2	0	2	long-unsigned	0x4308
Synchronization event counter	1	0-0:96.15.2.255	2	0	2	long-unsigned	763
Metrological firmware version	1	7-0:0.2.1.255	2	0	8	octet-string	"ABCDEFGH"
Metrological event counter	1	0-0:96.15.1.255	2	0	2	long-unsigned	1532
Non-metrological firmware version	1	7-1:0.2.1.255	2	0	8	octet-string	"DEFGHIJK"

Table 13 shows the attributes of the "Compact data" object.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 13 – Attributes of the “Compact data” object

capture_objects (array)	For the elements of the array, see columns 2, 3, 4 and 5 of Table 12.
template_id (unsigned)	1
template_description (octet-string)	-- For the data types, see column 7 of Table 12. 02 08 11 12 12 12 12 09 12 09
compact_buffer (octet-string) 29 bytes	-- For the values, see column 8 of Table 12. 01 4200 4108 4308 02FB 084142434445464748 05FC 084445464748494A4B

For comparison, the A-XDR encoding of the data as if they were read from the buffer attribute of a “Profile generic” object is shown in Table 14 (only the Data is shown).

Table 14 – Encoding the data read from the buffer attribute of a “Profile generic” object

Encoding	Explanation	Length
01 01	array of one element	2
02 07	structure of 7 elements	2
12 4200	long-unsigned	3
12 4108	long-unsigned	3
12 4308	long-unsigned	3
12 02FB	long-unsigned	3
09 08 4142434445464748	octet-string of length 8	10
12 05FC	long-unsigned	3
09 08 4445464748494A4B	octet-string of length 8	10
	Total	39 bytes

4.3.10.6.3 Logbook reading

In this example, the data to be compacted is the *buffer* attribute of a Logbook held by a “Profile generic” object capturing 2 elements, as shown in Table 15.

Table 15 – Logbook data

Data	class _id	Logical name	Attribute _id	data_ index	Size (bytes)	Type	Value
1	2	3	4	5	6	7	8
Unix time	1	0-0:1.1.0.255	2	0	4	double-long-unsigned	See Note
Event code	1	0-0:96.11.2.255	2	0	1	unsigned	

NOTE For this example, the following values are assumed:

– UNIX timestamp: 1374573317D,

– status: 0x29,

– for simplicity of the example, the values are the same for all entries,

– there are 50 entries in the buffer.

Table 16 shows the data to be captured by the “Compact data” object.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 16 – Attributes of the “Compact data” object

Data	class _id	Logical name	attribute _id	Data index	Size (bytes)	Type	Value
1	2	3	4	5	6	7	8
Template Id	62	0-0:66.0.0.255	4	0	1	unsigned	2
Logbook buffer[1]	7	0-0:99.98.0.255	2	0	dyn. [2]	array of structure	2

[1] See Table 15.

[2] The size is dynamic and depends on the number of entries captured.

Table 17 shows the attributes of the “Compact data” object.

Table 17 – Attributes of the “Compact data” object

capture_objects (array)	For the elements of the array, see columns 2, 3, 4 and 5 of Table 16. The capture object has only 2 elements: the Template_id and the *buffer* attribute of the Logbook.
template_id (unsigned)	2
template-description (octet-string)	-- For the data types, see column 7 of Table 16. 02 02 11 01 02 02 06 11 Meaning: 02 02 – a structure of 2 elements 11 – first element is an unsigned 01 02 02– second element is an array of structure with two elements in the structure 06 first one is a double-long-unsigned 11 second one is an unsigned
compact_buffer (octet-string)	-- For the values, see column 8 of Table 16. 02 -- value of the template-id 32 -- number of the elements in the array and 50*5 = 250 bytes (for 50 elements in the log book) 252 bytes in total

For comparison, the A-XDR encoding of the same data when read from the *buffer* attribute of a “Profile generic” object is shown in Table 18.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 18 – A-XDR encoding of the data read from the *buffer* attribute

Encoding	Explanation	Length
01 32	array of 50 elements	2
02 02	structure of 2 elements	2
06 51EE5305	double-long-unsigned	5
11 29	unsigned	2
02 02	structure of 2 elements	2
06 51EE5305	double-long-unsigned	5
11 29	unsigned	2
02 02	structure of 2 elements	2
06 51EE5305	double-long-unsigned	5
11 29	unsigned	2
….	…	…
	Total	**452 bytes**

4.4 Interface classes for access control and management

4.4.1 Overview

Interface classes in this category model the logical structure of the DLMS®/COSEM server, allow configuring and managing access to its resources, updating the firmware and managing security:

- the “Association SN” class – see 4.4.3 – and the “Association LN” class – see 4.4.4 – model AAs. Their instances, the Association objects provide the list of objects accessible in each AA, manage and control access rights to their attributes and methods. They also manage the authentication of the communicating partners;
- the “SAP Assignment” class – see 4.4.5 – models the logical structure of the server;
- the “Image transfer” class – see 4.4.6– models the firmware update process;
- the “Security setup” class – see 4.4.7 – models the elements of the security context. “Security setup” objects are referenced from the “Association” objects and allow configuring security suites and security policies and managing security material;
- the “Push setup” class – see 4.4.8 – models the push operation of the server;
- the “Data protection” class – see 4.4.9 – specifies the necessary elements to apply cryptographic protection to COSEM object attribute values as well as to method invocation and return parameters;
- the “Function control” class – see 4.4.10 – allows enabling and disabling functions in the server;
- the “Array manager” class – see 4.4.11 – allows managing attributes of type array of other interface objects.
- the “Communication port protection” class – see 4.4.12 – allows protection of ports against unauthorised attempts at communication.

4.4.2 Client user identification

This new feature enables the server to distinguish between different users from the client side and to log their activities accessing the meter.

Each AA established between a client and a server can be used by several users on the client side. The properties of the AA are configured in the server, using the “Association” and the “Security setup” objects. All users of an AA on the client side use these same properties.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The security keys are known by the client and the server but they need not be known by the users of the client.

The list of users – identified by their user_id and user_name – is known both by the client and the server. In the server it is held by the *user_list* attribute of the "Association" objects.

NOTE The way a client authenticates a user to log into a client system is outside the scope of this specification.

During AA establishment, the user_id – belonging to the user_name – is carried by the calling-AE-invocation-id field of the AARQ APDU. If the user_id provided is on the *user_list*, the AA can be established – provided that all other conditions are met – and the *current_user* attribute is updated. The value of this attribute can be logged.

If the server does not "know" the user, the AA shall not be established. The server may silently discard the request to establish the AA or it may send back an appropriate error message.

The user identification process is optional: if the *user_list* is empty – i.e. it is an array of 0 elements – the function is disabled.

4.4.3 Association SN (class_id = 12, version = 4)

4.4.3.1 Overview

COSEM logical devices able to establish AAs within a COSEM context using SN referencing, model the AAs using instances of the "Association SN" IC. A COSEM logical device may have one instance of this IC for each AA the device is able to support.

The **short_name** of the "Association SN" object itself is fixed within the COSEM context. See 4.1.3.

Association SN	**0...n**	**class_id = 12, version = 4**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. object_list (static)	objlist_type				x + 0x08
3. access_rights_list (static)	access_rights_type				x + 0x10
4. security_setup_reference (static)	octet-string				x + 0x18
5. user_list (static)	array				x + 0x20
6. current_user	structure				x + 0x28
Specific methods	***m/o***				
1. *reserved from previous versions*	o				
2. *reserved from previous versions*	o				
3. read_by_logicalname (data)	o				x + 0x30
4. *reserved from previous versions*	o				
5. change_secret (data)	o				x + 0x40
6. *reserved from previous versions*	o				
7. *reserved from previous versions*					
8. reply_to_HLS_authentication (data)	o				x + 0x58
9. add_user (data)	o				x + 0x60
10. remove_user (data)	o				x + 0x68

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.3.2 Attribute description

4.4.3.2.1 logical_name

Identifies the “Association SN” object instance. See 6.2.33.

4.4.3.2.2 object_list

Contains the list of all objects with their base_name (short_name), class_id, version and *logical_name*. The base_name is the DLMS® objectName of the first attribute (*logical_name).*

```
objlist_type::= array          objlist_element

objlist_element::= structure
{
      base_name:      long,
      class_id:       long-unsigned,
      version:        unsigned,
      logical_name:   octet-string
}
```

selective access (see 4.1.4.17) to the attribute *object_list* may be available. The access selector values and their parameters are as defined in Table 19.

4.4.3.2.3 access_rights_list

Contains the access rights to attributes and methods.

The link between the *object_list* and the *access_rights_list* is the base_name, present in both the objlist_element structure and the access_right_element structure. Therefore, the base_names on the two lists shall be the same. The number – and preferably, the order – of the elements in the array of objlist_element and the array of access_right_element shall also be the same.

```
access_rights_type::= array          access_rights_element

access_rights_element::= structure
{
      base_name:              long,
      attribute_access:       attribute_access_descriptor,
      method_access:          method_access_descriptor
}

attribute_access_descriptor::= array      attribute_access_item

attribute_access_item::= structure
{
      attribute_id:           integer,
      access_mode:            enum,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

When the enum value is interpreted as an unsigned8, the meaning of each bit is as shown below:

```
                    {
                        Bit     attribute access_mode

                        (0)     read-access,
                        (1)     write-access,
                        (2)     authenticated request,
                        (3)     encrypted request,
                        (4)     digitally signed request,
                        (5)     authenticated response,
                        (6)     encrypted response,
                        (7)     digitally signed response
                    }


    access_selectors:   CHOICE
                    {
                        null-data           [0],
                        array integer       [1]
                    }
                }
method_access_descriptor::= array       method_access_item

                method_access_item::= structure
                {
                    method_id:          integer,
                    access_mode:        enum
```

When the enum value is interpreted as an unsigned8, the meaning of each bit is as shown below:

```
                    {
                        Bit     method access_mode

                        (0)     access,
                        (1)     not-used,
                        (2)     authenticated request,
                        (3)     encrypted request,
                        (4)     digitally signed request,
                        (5)     authenticated response,
                        (6)     encrypted response,
                        (7)     digitally signed response
                    }
                }
```

selective access (see 4.1.4.17) to the attribute access_rights_list may be available (optional). The access selector values and their parameters are as defined in Table 19.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 19 – Parameters for selective access to the object_list and access_rights_list attribute

Access selector value	Parameter	Available with attribute	Comment
1	class_id: long-unsigned	2	Delivers the subset of the object_list for a specific class_id. For the response: data::= objlist_type

2	structure
	{
	class_id: long-unsigned,
	logical_name: octet-string
	}
		2	Delivers the entry of the object_list for a specific class_id and *logical_name*.
			For the response: data::= objlist_element

3	base_name: long	2, 3	In the case of attribute 2, delivers the entry of the object_list for a specific base_name. For the response: data::= objlist_element In the case of attribute 3, delivers the entry of the access_rights_list for a specific base_name. For the response: data::=access_rights_element

4.4.3.2.4 security_setup_reference

References the “Security setup” object by its *logical_name*. The referenced object manages security for a given “Association SN” object instance.

4.4.3.2.5 user_list

Contains the list of users allowed to use the AA managed by the given instance of the “Association SN” IC.

```
array          user_list_entry

user_list_entry::=      structure
{
    user_id:            unsigned,

    user_name           CHOICE
                        {
                        visible-string      [10],
                        UTF-8-string        [12]
                        }
}
```

Where:

– user_id is the identifier of the user (this value is carried by the calling-AE-invocation-id field of the AARQ);

– user_name is the name of the user.

If the user_list attribute is empty – i.e. it is an array of 0 elements – any user can use the AA, i.e. the calling-AE-invocation-id field of the AARQ is ignored.

If the user_list attribute is not empty then only the users in the list can establish the AA, i.e. the calling-AE-invocation-id field of the AARQ shall be present and its value shall match one of user_ids in the user_list or else, the AA is not established.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.3.2.6 current_user

Holds the identifier of the current user.

current_user::= user_list_entry (see 4.4.3.2.5)

If the user_list is empty, then current_user shall be a structure {user_id: unsigned 0, user_name: visible string of 0 elements}.

4.4.3.3 Method description

4.4.3.3.1 read_by_logicalname (data)

Reads attributes for selected objects. The objects are specified by their class_id and their *logical_name*. With this method, the parameterized access feature can also be used.

```
data::= array       attribute_identification

attribute_identification::=     structure
{
        class_id:               long-unsigned,
        logical_name:           octet-string,
        attribute_index:        integer
}
```

where attribute_index is a pointer (i.e. offset) to the attribute within the object.

attribute_index 0 delivers all attributes; attribute_index 1 delivers the first attribute (i.e. *logical_name*), etc.).

For the response: data is according to the type of the attribute.

NOTE If at least one attribute has no read access right under the current association, then a *read_by_logicalname* () to attribute index 0 reveals the error message “scope-of-access-violated”, see IEC 62056-5-3:2023, Clause 8.

4.4.3.3.2 change_secret (data)

Changes the LLS or HLS secret (for example password).

data::= octet-string new secret

NOTE 1 The structure of the “new secret” depends on the security mechanism implemented. The “new secret” may contain additional check bits and it may be encrypted.

NOTE 2 In the case of HLS with GMAC, the (HLS_) secret is held by the “Security setup“ object referenced in attribute.

4.4.3.3.3 reply_to_HLS_authentication (data)

The remote invocation of this method delivers to the server the result of the secret processing by the client of the server’s challenge to the client, f(StoC), as the data service parameter of the Read.request primitive invoked with parameterised access.

data::= octet-string client’s response to the challenge

If the authentication is accepted, then the response (Read.confirm primitive) contains Result == OK and the result of the secret processing by the server of the client’s challenge to the server, f(CtoS) in the data service parameter of the Read.response service.

data::= octet-string server's response to the challenge

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

4.4.3.3.4 add_user (data)

Adds a user to the user_list.

data::= user_list_entry (see 4.4.3.2.5)

4.4.3.3.5 remove_user (data)

Removes a user from the user_list.

data::= user_list_entry (see 4.4.3.2.5)

4.4.4 Association LN (class_id = 15, version = 3)

4.4.4.1 Overview

COSEM logical devices able to establish AAs within a COSEM context using LN referencing, model the AAs through instances of the “Association LN” IC. A COSEM logical device has one instance of this IC for each AA the device is able to support.

Association LN		**0...MaxNbofAss.**	**class_id = 15, version = 3**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. object_list	(static)	object_list_type				x + 0x08
3. associated_partners_id		associated_partners_type				x + 0x10
4. application_context_name		context_name_type				x + 0x18
5. xDLMS_context_info		xDLMS_context_type				x + 0x20
6. authentication_mechanism_name		mechanism_name_type				x + 0x28
7. secret		octet-string				x + 0x30
8. association_status		enum				x + 0x38
9. security_setup_reference	(static)	octet-string				x + 0x40
10. user_list	(static)	array				x + 0x48
11. current_user		structure				x + 0x50
Specific methods		***m/o***				
1. reply_to_HLS_authentication (data)		o				x + 0x60
2. change_HLS_secret (data)		o				x + 0x68
3. add_object (data)		o				x + 0x70
4. remove_object (data)		o				x + 0x78
5. add_user (data)		o				x + 0x80
6. remove_user (data)		o				x + 0x88

4.4.4.2 Attribute description

4.4.4.2.1 logical_name

Identifies the “Association LN” object instance. See 6.2.33.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.2 object_list

4.4.4.2.2.1 General

Contains the list of visible COSEM objects with their class_id, version, *logical_ name* and the access rights to their attributes and methods within the given AA.

```
object_list_type::= array              object_list_element
object_list_element::= structure

{
      class_id:              long-unsigned,
      version:               unsigned,
      logical_name:          octet-string,
      access_rights:         access_right
}

access_right::= structure
{
      attribute_access:      attribute_access_descriptor,
      method_access:         method_access_descriptor
}

attribute_access_descriptor::= array      attribute_access_item

attribute_access_item::=      structure
{
      attribute_id:          integer,
      access_mode:           enum
```

When the enum value is interpreted as an unsigned8, the meaning of each bit is as shown below:

```
{
            Bit    attribute access_mode

            (0)    read-access,
            (1)    write-access,
            (2)    authenticated request,
            (3)    encrypted request,
            (4)    digitally signed request,
            (5)    authenticated response,
            (6)    encrypted response,
            (7)    digitally signed response
      }

      access_selectors:      CHOICE
      {
            null-data          [0],
            array integer      [1]
      }
}

method_access_descriptor::= arraymethod_access_item

method_access_item::= structure
{
      method_id:       integer,
      access_mode:     enum

      When the enum value is interpreted as an unsigned8, the meaning of each
      bit is as shown below:
      {
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
                    Bit     method access_mode

                    (0)     access,
                    (1)     not-used,
                    (2)     authenticated request,
                    (3)     encrypted request,
                    (4)     digitally signed request,
                    (5)     authenticated response,
                    (6)     encrypted response,
                    (7)     digitally signed response
                }
            }
```

Where:

- the attribute_access_descriptor and the method_access_descriptor elements always contain all implemented attributes or methods;
- access_selectors contain a list of the supported selector values.

selective access (see 4.1.4.17) to the attribute *object_list* may be available (optional). The selective access parameters are as defined in 4.4.4.2.2.2.

4.4.4.2.2.2 Parameters for selective access to the *object_list* attribute

- If no selective access is requested, (no Access_Selection_Parameters parameter is present in the GET.request (.indication) service primitive for the object_list attribute) the corresponding .response (.confirmation) service shall contain all object_list_elements of the object_list attribute.
- When selective access is requested to the object_list attribute (the Access_Selection_Parameter parameter is present), the response shall contain a 'filtered' list of object_list_elements, as shown in Table 20:

Table 20 – Parameters for selective access to the object_list attribute

Access selector	Access parameter	Comment
1	NULL	All information excluding the access_rights shall be included in the response.
2	class_list	Access by class_id. In this case, only those object_list_elements of the *object_list* shall be included in the response, which have a class_id equal to one of the class_id-s of the class_list. No access_right information is included. class_list::= array class_id class_id: long-unsigned

3	object_id_list	Access by object. The full information record of object instances on the object_id_list shall be returned.
		object_id_list::= array object_id
		object_id::= structure
		{
		class_id: long-unsigned,
		logical_name: octet-string
		}

4	object_id	The full information record of the required COSEM object instance shall be returned. object_id::= structure See above.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.3 associated_partners_id

Contains the identifiers of the COSEM client and server (logical device) APs within the physical devices hosting these APs, which belong to the AA modelled by the “Association LN” object.

```
associated_partners_type::= structure
{
client_SAP:       integer,
server_SAP:       long-unsigned
}
```

The range for the client_SAP is 0…0x7F.

The range for the server_SAP is 0x0000…0x3FFF.

The SAPs shall be in the range allowed by the data type and the media.

4.4.4.2.4 application_context_name

In the COSEM environment, it is intended that an application context pre-exists and is referenced by its name during the establishment of an AA. This attribute contains the name of the application context for that AA.

```
context_name_type::=          CHOICE
{
      context_name_structure          [2],
      octet-string                    [9]
}
```

The application context name is specified as OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.2.

When the context_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
context_name_structure::= structure
{
      joint_iso_ctt_element:                unsigned,
      country_element:                      unsigned,
      country_name_element:                 long-unsigned,
      identified_organization_element:      unsigned,
      DLMS_UA_element:                      unsigned,
      application_context_element:          unsigned,
      context_id_element:                   unsigned
}
```

EXAMPLE 1: In the case of context_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 01 11 01 (all values are hexadecimal).

When the context_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

EXAMPLE 2: In the case of context_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 01 01 (all values are hexadecimal).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.5 xDLMS_context_info

Contains all the necessary information on the xDLMS context for the given AA.

```
xDLMS_context_type::= structure
{
    conformance:                bit-string,
    max_receive_pdu_size:       long-unsigned,
    max_send_pdu_size:          long-unsigned,
    dlms_version_number:        unsigned,
    quality_of_service:         integer,
    cyphering_info:             octet-string
}
```

Where:

– the conformance element contains the xDLMS conformance block supported by the server. The length of the bit-string is 24 bits;

– the max_receive_pdu_size element contains the maximum length for an xDLMS APDU, expressed in bytes that the client may send negotiated during the application association process and limited by the server-max-receive-pdu-size parameter of the xDLMS initiateResponse APDU;

– the max_send_pdu_size element, in an active AA contains the maximum length for an xDLMS APDU, expressed in bytes that the server may send. It is limited by the client-max-receive-pdu-size parameter of the xDLMS initiateRequest APDU;

– the dlms_version_number element contains the DLMS® version number supported by the server;

– the quality_of_service element is not used;

– the cyphering_info element – in an active AA – contains the dedicated key parameter of the xDLMS initiateRequest APDU. See IEC 62056-5-3:2023, Clause 8.

4.4.4.2.6 authentication_mechanism_name

Contains the name of the authentication mechanism for the AA.

```
mechanism_name_type::=    CHOICE
{
    mechanism_name_structure        [2],
    octet-string                    [9]
}
```

The authentication mechanism name is specified as an OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.3.

When the mechanism_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
mechanism_name_structure::= structure
{
    joint_iso_ctt_element:                  unsigned,
    country_element:                        unsigned,
    country_name_element:                   long-unsigned,
    identified_organization_element:        unsigned,
    DLMS_UA_element:                        unsigned,
    authentication_mechanism_name_element:  unsigned,
    mechanism_id_element:                   unsigned
}
```

EXAMPLE 3: In the case of mechanism_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 02 11 01 (all values are hexadecimal):

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

When the mechanism_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

EXAMPLE 4: In the case of mechanism_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 02 01 (all values are hexadecimal).

No mechanism_name is required when no authentication is used

4.4.4.2.7 secret

Contains the secret for the LLS or HLS authentication process.

NOTE In the case of HLS with GMAC, the (HLS) secret is held by the "Security setup" object referenced in attribute 9, *security_setup_ reference, 4.4.4.2.9*.

4.4.4.2.8 association_status

Indicates the current status of the association, which is modelled by the object.

enum:

(0) non-associated,association-pending,
(1) associated

4.4.4.2.9 security_setup_ reference

References a "Security setup" object by its logical name. The referenced object manages security for a given "Association LN" object instance.

4.4.4.2.10 user_list

Contains the list of users allowed to use the AA managed by the given instance of the "Association LN" IC.

```
array          user_list_entry

user_list_entry::=      structure
{
    user_id:                unsigned,

    user_name       CHOICE
                    {
                    visible-string      [10],
                    UTF-8-string        [12]
                    }
}
```

Where:

user_id is the identifier of the user (this value is carried by the calling-AE-invocation-id field of the AARQ);

user_name is the name of the user.

If the *user_list* attribute is empty – i.e. it is an array of 0 elements – any user can use the AA, i.e. the calling-AE-invocation-id field of the AARQ is ignored.

If the *user_list* attribute is not empty then only the users in the list can establish the AA, i.e. the calling-AE-invocation-id field of the AARQ shall be present and its value shall match one of user_ids in the *user_list* or else the AA is not established.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.11 current_user

Holds the identifier of the current user.

current_user::= user_list_entry (see 4.4.4.2.10)

If the *user_list* is empty, then current_user shall be a structure {user_id: unsigned 0, user_name: visible string of 0 elements}.

4.4.4.3 Method description

4.4.4.3.1 reply_to_HLS_authentication (data)

The remote invocation of this method delivers to the server the result of the secret processing by the client of the server's challenge to the client, f(StoC), as the *data* service parameter of the ACTION.request primitive invoked.

data::= octet-string client's response to the challenge

If the authentication is accepted, then the response (ACTION.confirm primitive) contains Result == OK and the result of the secret processing by the server of the client's challenge to the server, f(CtoS) in the *data* service parameter of the response service.

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

4.4.4.3.2 change_HLS_secret (data)

Changes the HLS secret (for example encryption key).

data::= octet-string new HLS secret

The structure of the "new secret" depends on the security mechanism implemented. The "new secret" may contain additional check bits and it may be encrypted.

4.4.4.3.3 add_object (data)

Adds the referenced object to the *object_list*.

data::= object_list_element (see 4.4.4.2.2)

4.4.4.3.4 remove_object (data)

Removes the referenced object from the *object_list*.

data::= object_list_element (see 4.4.4.2.2)

4.4.4.3.5 add_user (data)

Adds a user to the *user_list*.

data::= user_list_entry (see 4.4.4.2.10)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.3.6 remove_user (data)

Removes a user from the *user_list*.

data::= user_list_entry (see 4.4.4.2.10)

4.4.5 SAP assignment (class_id = 17, version = 0)

4.4.5.1 Overview

This IC allows modelling the logical structure of physical devices, by providing information on the assignment of the logical devices to their SAPs. See IEC 62056-5-3:2023, Annex A.

SAP assignment	**0...1**	**class_id = 17, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. SAP_assignment_list (static)	asslist_type			0	x + 0x08
Specific methods	***m/o***				
1. connect_logical_device (data)	o				x + 0x20

4.4.5.2 Attribute description

4.4.5.2.1 logical_name

Identifies the “SAP assignment” objects instance. See 6.2.34.

4.4.5.2.2 SAP_assignment_ list

Contains the list of all logical devices and their SAP addresses within the physical device.

```
asslist_type::= array          asslist_element
asslist_element::= structure
{
        SAP:                   long-unsigned,
        logical_device_name:   CHOICE
                {
                octet-string       [9],
                visible-string     [10],
                utf8-string        [12]
                }
}
```

REMARK: The actual addressing is performed by the supporting communication layers

4.4.5.3 Method – connect_logical_ device (data)

Connects a logical device to a SAP. Connecting to SAP 0 will disconnect the device. More than one device cannot be connected to one SAP (exception SAP 0).

data::= asslist_element

4.4.6 Image transfer (class_id = 18, version = 0)

4.4.6.1 General

Instances of the Image transfer IC model the process of transferring binary files, called Images to COSEM servers.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE This specification includes some improvements and precisions to the text. The main changes are:

- The description of the Image transfer process is given as an example only. The text and the flow chart are updated;
- Data exchange between the client and a conceptual Image server is out of Scope;
- A clear distinction is made between the Image transferred and the Images to activate;
- Steps 1 and 6 are optional now;
- Size of the *image_transferred_blocks_status* bit-string may be dynamic;
- It is specified now that setting the value of the *image_transfer_enabled* attribute to FALSE disables the image transfer process;
- It is specified now that re-initiating the image transfer process resets the whole process;
- Some precisions have been added to the effect of invoking the methods;

4.4.6.2 The steps of the image transfer process

The Image transfer usually takes place in several steps:

- Step 1: (Optional): Get ImageBlockSize;
- Step 2: Client initiates Image transfer;
- Step 3: Client transfers ImageBlocks;
- Step 4: Client checks completeness of the Image;
- Step 5: Server verifies the Image (Initiated by the client or on its own);
- Step 6 (Optional): Client checks the information on the images to activate;
- Step 7: Server activates the Image(s) (Initiated by the client or on its own).

For an example with more detailed explanations, see 4.4.6.6.

4.4.6.3 Overview

This IC allows provides the attributes and methods with which to model the image transfer process.

Image transfer		**0...n**	**class_id = 18, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. image_block_size	(static)	double-long-unsigned				x + 0x08
3. image_transferred_blocks_status	(dyn.)	bit-string				x + 0x10
4. image_first_not_transferred_block_number	(dyn.)	double-long-unsigned				x + 0x18
5. image_transfer_enabled	(static)	boolean				x + 0x20
6. image_transfer_status	(dyn.)	enum				x + 0x28
7. image_to_activate_info	(dyn.)	array				x + 0x30
Specific methods		***m/o***				
1. image_transfer_initiate (data)		m				x + 0x40
2. image_block_transfer (data)		m				x + 0x48
3. image_verify (data)		m				x + 0x50
4. image_activate (data)		m				x + 0x58

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.4 Attribute description

4.4.6.4.1 logical_name

Identifies the “Image transfer” object instance. See 6.2.37.

4.4.6.4.2 image_block_size

Holds the ImageBlockSize, expressed in octets, which can be handled by the server. ImageBlockSize shall not exceed the ServerMaxReceivePduSize negotiated.

NOTE *image_block_size* is a property of the server.

4.4.6.4.3 image_transferred_blocks_status

Provides information about the transfer status of each ImageBlock. Each bit in the bit-string provides information about one individual ImageBlock:

0 = Not transferred,

1 = Transferred

NOTE The size of the attribute may be dynamic, i.e. it may grow upon reception of new ImageBlocks.

4.4.6.4.4 image_first_not_transferred_block_number

Provides the ImageBlockNumber of the first ImageBlock not transferred. Once the Image is complete, the value returned should be equal to or above the number of blocks calculated from the Image size and the ImageBlockSize.

4.4.6.4.5 image_transfer_enabled

Controls the enabling of the Image transfer process.

boolean:

FALSE = Disabled,
TRUE = Enabled

The image transfer methods can be invoked successfully only if the value of this attribute is TRUE. Setting the value of this this attribute to FALSE disables all methods (invoking them fails).

4.4.6.4.6 image_transfer_status

Holds the status of the Image transfer process.

enum:

(0) Image transfer not initiated,
(1) Image transfer initiated,
(2) Image verification initiated,
(3) Image verification successful,
(4) Image verification failed,
(5) Image activation initiated,
(6) Image activation successful,
(7) Image activation failed

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.4.7 image_to_activate_info

Provides information on the Image(s) ready for activation. It is generated as the result of the Image verification process. The client may check this information before activating the Images.

```
array        image_to_activate_info_element

image_to_activate_info_element::= structure
{
image_to_activate_size:                 double-long-unsigned,
image_to_activate_identification:       octet-string,
image_to_activate_signature:            octet-string
}
```

Where:

image_to_activate_size	is the size of the Image to be activated, expressed in octets;
image_to_activate_identification	is the identification of the Image to be activated, and may contain information like manufacturer, device type, version information, etc.;
image_to_activate_signature	is the signature of the Image to be activated.

NOTE The algorithm to generate the signature is out of the Scope of this specification.

4.4.6.5 Method description

4.4.6.5.1 image_transfer_initiate (data)

Initializes the Image transfer process.

```
data::= structure
{
    image_identifier:       octet-string,
    image_size:             double-long-unsigned
}
```

Where:

– image_identifier identifies the Image to be transferred;

– image_size holds the ImageSize, expressed in octets.

NOTE The *image_identifier* identifies the Image to be transferred (container), but it is not necessarily linked to its content, i.e. the Images which will be activated. That information can be retrieved from the *image_to_activate* attribute after verification of the Image transferred.

After a successful invocation of the method the *image_transfer_status* attribute is set to (1) and the *image_first_not_transferred_block_number* is set to 0. Any subsequent invocation of the method resets the whole Image transfer process and all ImageBlocks need to be transferred again.

4.4.6.5.2 image_block_transfer (data)

Transfers one block of the Image to the server.

```
data::= structure
{
    image_block_number:     double-long-unsigned,
    image_block_value:      octet-string
}s
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

After a successful invocation of the method the corresponding bit in the *image_transferred_blocks_status* attribute is set to 1 and the *image_first_not_transferred_block_number* attribute is updated.

4.4.6.5.3 image_verify (data)

Verifies the integrity of the Image before activation.

data::= integer (0)

The result of the invocation of this method may be success, temporary_failure or other_reason. If it is not success, then the result of the verification can be found by retrieving the value of the *image_transfer_status* attribute.

In the case of success, the *image_to_activate_info* attribute holds the information about the images to activate.

NOTE xDLMS services Action-Result / Data-Access-Result codes are specified in IEC 62056-5-3:2023, Clause 8.

4.4.6.5.4 image_activate (data)

Activates the Image.

data::= integer (0)

If the Image transferred has not been verified before, then this is done as part of the Image activation.

The result of the invocation of this method may be success, temporary-failure or other-reason. If it is not success, then the result of the activation can be learned by retrieving the value of the *image_transfer_status* attribute.

NOTE xDLMS services Action-Result / Data-Access-Result codes are specified in IEC 62056-5-3:2023, Clause 8.

4.4.6.6 Example for a standard image transfer process

4.4.6.6.1 General

In this subclause an example of a usual Image transfer process from a client perspective is given including a flow chart shown in Figure 11. Please note that it is not mandatory to follow the process; depending on the use case some steps may be omitted or others may be necessary.

4.4.6.6.2 Precondition:

The Image transfer has to be enabled: *image_transfer_enabled* = TRUE.

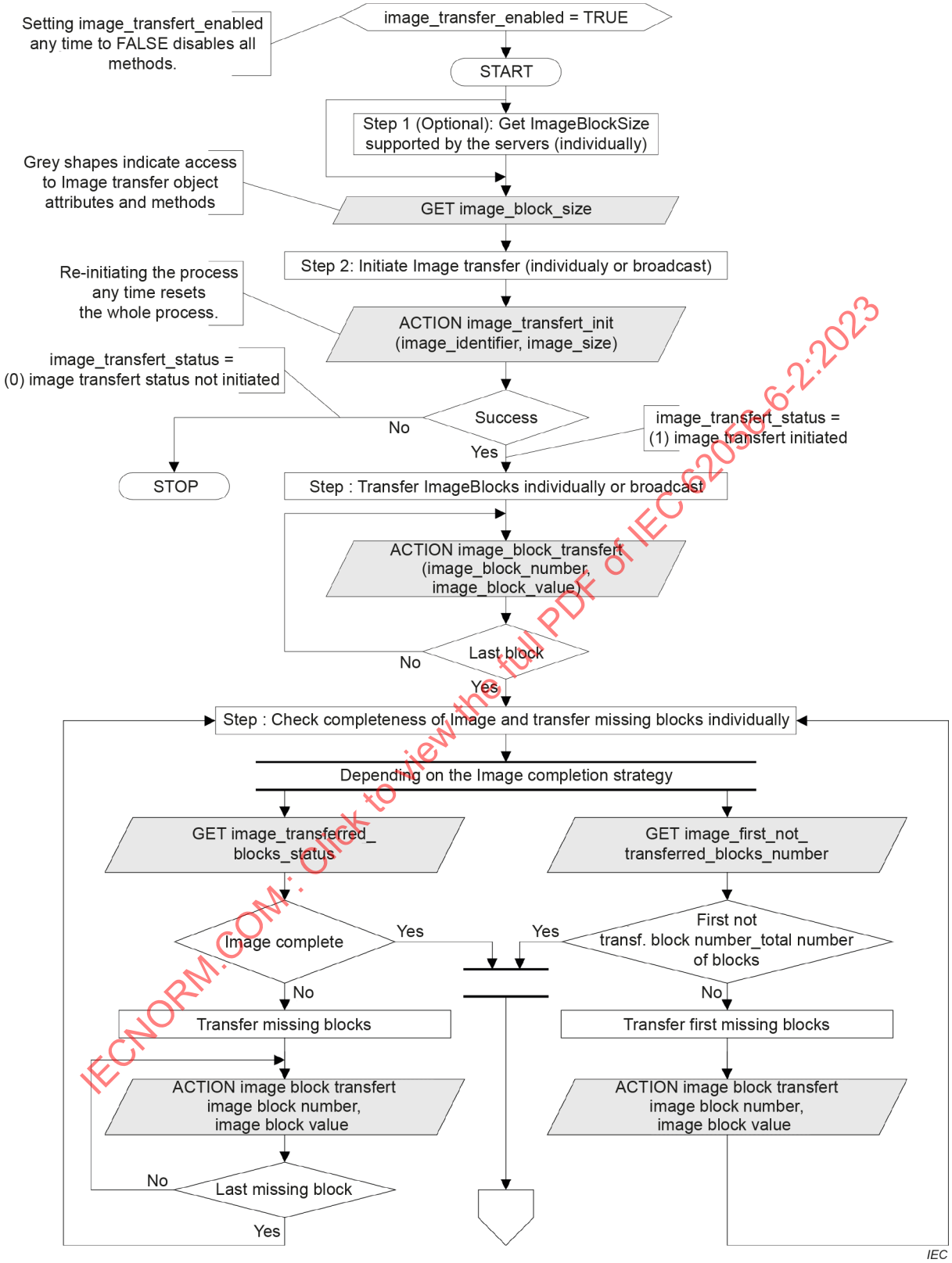
Setting *image_transfer_enabled* any time to FALSE disables all methods (invoking those methods fails). The value of the status attributes is not defined.

4.4.6.6.3 Step 1 (Optional): Get ImageBlockSize

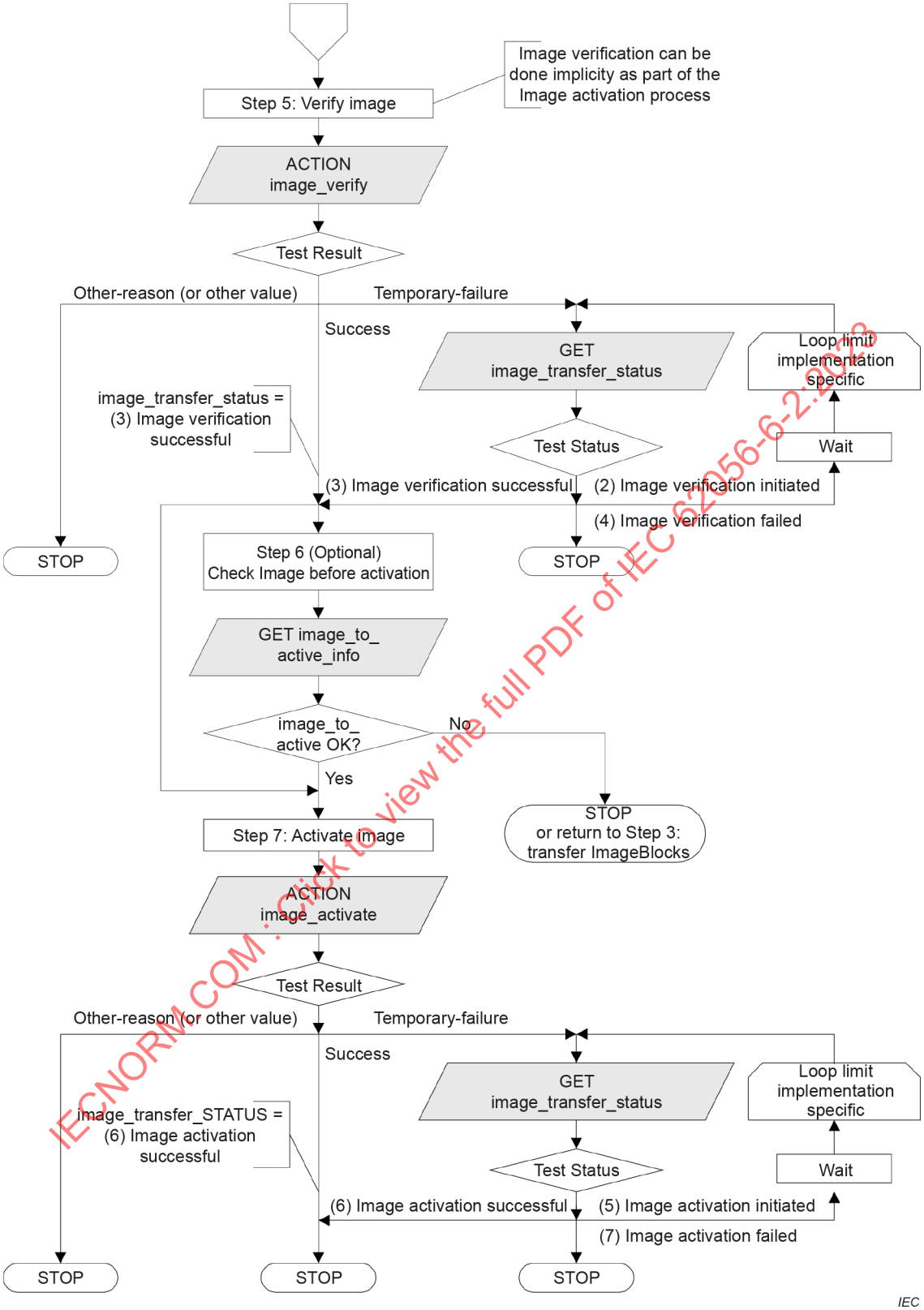
If the client does not know the size of the image blocks the image transfer target server can handle, it shall read the *image_block_size* attribute of the relevant “Image transfer” object of each server the image has to be transferred to, before starting the process. The client can transfer then ImageBlocks of the right size.

If ImageBlocks are sent using broadcast to a group of COSEM servers the ImageBlockSize shall be the same in each member of the group.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 11 – Image transfer process flow chart

4.4.6.6.4 Step 2: Client initiates Image transfer

The client initiates the Image transfer process individually or using broadcast in all servers by invoking the *image_transfer_initiate* method. The method invocation parameter holds the identifier and the size of the Image to be transferred. The server shall make the memory space – necessary to accommodate the Image – available.

After a successful initiation, the value of the *image_transfer_status* attribute is (1). The *image_transferred_blocks_status* attribute shall be reset, the value of the *image_first_not_transferred_block_number* attribute shall be set to 0 and the value of the *image_to_activate_info* attribute should be reset. The image transfer process is initiated and the COSEM server is prepared to accept ImageBlocks.

4.4.6.6.5 Step 3: Client transfers ImageBlocks

The client transfers ImageBlocks to (a group of) server(s) by invoking the *image_block_transfer* method individually or using broadcast. The method invocation parameters include the ImageBlockNumber and one ImageBlock. ImageBlocks are accepted only by those COSEM servers, in which the Image transfer process has been successfully initiated. Other servers silently discard any ImageBlocks received.

4.4.6.6.6 Step 4: Client checks completeness of the Image

The client checks – with each server individually – the completeness of the Image transferred. If the Image is not complete, it transfers the ImageBlocks not (yet) transferred. This is an iterative process, continued until the whole Image is successfully transferred.

To identify and transfer the ImageBlocks not transferred, two mechanisms are available.

- the client may retrieve the status of each ImageBlock: either not transferred or transferred. This is performed by retrieving the value of the *image_transferred_blocks_status* attribute. The client transfers then the ImageBlocks not (yet) transferred;
- alternatively, the client may retrieve the ImageBlockNumber of the first block not transferred. This is performed by retrieving the value of the *image_first_not_transferred_block_number* attribute. The client transfers then this ImageBlock not (yet) transferred;
- after this, the client checks again he completeness of the Image.

NOTE The two mechanisms can be freely combined.

4.4.6.6.7 Step 5: Server verifies the Image

The Image is verified by the server. This can be initiated by invoking the *image_verify* method by the client or it may be initiated also by the server. The result can be:

- success, if the verification could be completed;
- temporary-failure, if the verification has not been completed;
- other-reason, if the verification failed.

NOTE The conditions of verifying the Image are out of the scope of this document.

4.4.6.6.8 Step 6 (Optional): Client checks the information on the images to activate

The result of the Image verification may be checked by the client by retrieving the value of the *image_transfer_status* attribute. The value of this attribute is updated as a result of the Image verification.

An Image transferred may hold one or more Images to be activated. For each Image to be activated, the image_to_activate_info attribute holds the parameters: {image_to_activate_size, image_to_activate_identification, image_to_activate_signature}.

If this information is not what is expected, the client can restart transferring the image.

Otherwise it goes to the next step, activation of the Image.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.6.9 Step 7: Server activates the Image(s)

The Image is activated by the server. This can be initiated by invoking the *image_activate* method by the client or it may be initiated also by the server. If the activation is done without a previous verification, then verification is done implicitly as part of the activation. The result of the invocation of the *Image_activate* method can be:

- success, if the Image activation has been successfully started;
- temporary-failure, if the verification/activation has not been completed;
- other-reason, if the activation failed.

In the case of success, the server performs the activation of the new Image(s). During this process, it is not accessible. After the Image(s) has (have) been activated, the result of may be checked by the client by retrieving the value of the *image_transfer_status* attribute or by reading the contents of the appropriate COSEM objects holding the identifier, version and digital signature of the active firmware.

4.4.7 Security setup (class_id = 64, version = 1)

4.4.7.1 Overview

Instances of the “Security setup” IC contain the necessary information on the security suite in use and the security policy applicable between a client and a server identified by their respective system title. They also provide methods to increase the level of security and to manage symmetric keys, asymmetric key pairs and certificates.

Security setup	**0…n**	**class_id = 64, version = 1**			
Attributes	**Data type**	**Min.**	**Max.**	**Def.**	**Short name**
1. logical_name (static)	octet-string				x
2. security_policy (static)	enum				x + 0x08
3. security_suite (static)	enum				x + 0x10
4. client_system_title (dyn.)	octet-string				x + 0x18
5. server_system_title (static)	octet-string				x + 0x20
6. certificates (dyn.)	array				x + 0x28
Specific methods	m/o				
1. security_activate (data)	o				x + 0x30
2. key_transfer (data)	o				x + 0x38
3. key_agreement (data)	o				x + 0x48
4. generate_key_pair (data)	o				x + 0x50
5. generate_certificate_request (data)	o				x + 0x58
6. import_certificate (data)	o				x + 0x60
7. export_certificate (data)	o				x + 0x68
8. remove_certificate (data)	o				x + 0x70

4.4.7.2 Attribute description

4.4.7.2.1 logical_name

Identifies the “Security setup” object instance. See 6.2.36.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.7.2.2 security_policy

Enforces authentication and/or encryption and/or digital signature using the security algorithms available within the security suite.

It applies independently for requests and responses.

enum: When the enum value is interpreted as an unsigned, the meaning of each bit is as shown below:

Bit	Security policy
0	unused, shall be set to 0,
1	unused, shall be set to 0,
2	authenticated request,
3	encrypted request,
4	digitally signed request,
5	authenticated response,
6	encrypted response,
7	digitally signed response

NOTE With this, the value (0) means that no cryptographic protection is required, as in version 0 of the “Security setup” IC.

The access rights may require stronger protection than what is required by the security policy.

4.4.7.2.3 security_suite

Specifies the security algorithms available.

enum:

(0) AES-GCM-128 authenticated encryption and AES-128 key wrap,
(1) AES-GCM-128 authenticated encryption, ECDSA P-256 digital signature, ECDH P-256 key agreement, SHA-256 hash, V.44 compression and AES-128 key wrap,
(2) AES-GCM-256 authenticated encryption, ECDSA P-384 digital signature, ECDH P-384 key agreement, SHA-384 hash, V.44 compression and AES-256 key wrap,
(3) … (15) reserved

4.4.7.2.4 client_system_title

Carries the (current) client system title:

– in the S-FSK PLC environment, the active initiator sends its system title using the CIASE protocol;

NOTE It is also held by the active_initiator attribute of the S-FSK Active initiator object; see 4.10.4.

– during confirmed or unconfirmed AA establishment, it is carried by the calling-AP-title field of the AARQ APDU;

If a client system title has already been sent during a registration process, like in the case of the S-FSK PLC profile, the client system title carried the AARQ APDU should be the same. Otherwise, the AA shall be rejected and appropriate diagnostic information should be sent.

– in a pre-established AA, it can be written by the client using an unsecured service.

4.4.7.2.5 server_system_title

Carries the server system title.

– in the S-FSK PLC environment, the server sends its system title during the discover process, using the CIASE protocol;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– during confirmed AA establishment, it is carried by the responding-AP-title field of the AARE APDU.

This attribute shall be read only.

4.4.7.2.6 certificates

Carries information on X.509 v3 certificates available and stored in the server.

```
array                certificate_info

certificate_info::= structure
{
        certificate_entity: enum:
                                    (0) server,
                                    (1) client,
                                    (2) certification authority,
                                    (3) other
        certificate_type: enum:
                                    (0) digital signature,
                                    (1) key agreement,
                                    (2) TLS,
                                    (3) other
        serial_number:              octet-string,
        issuer:                     octet-string,
        subject:                    octet-string,
        subject_alt_name:           octet-string
}
```

For the elements serial_number, issuer, subject, subject_alt_name see IEC 62056-5-3:2023, 5.6.4.

A server that uses public key cryptography stores the following public key certificates:

For the server:

– Digital Signature Certificate,
– Static Key Agreement Certificate,
– TLS Certificate.

For each client and third party:

– Digital Signature Certificate,
– Static Key Agreement Certificate,
– TLS Certificate.

For Certification Authorities:

– Certification Authority Certificate.

4.4.7.3 Method description

4.4.7.3.1 security_activate (data)

Activates and strengthens the security policy:

data::= enum, as specified at the *security_policy* attribute.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Strengthening the security policy means that another kind of protection is added. Once a kind of protection is added, it cannot be removed.

Once a kind of protection is added, it cannot be removed. If the method is invoked with a value indicating a security policy that is weaker than the value of the security_policy attribute, the method invocation fails.

The new security policy applies as soon as the method invocation has been confirmed with success.

4.4.7.3.2 key_transfer (data)

Used to transfer one or more symmetric keys.

The *data* parameter includes identification of the key(s) transported and the wrapped key(s) themselves.

```
data::=       array key_transfer_data

key_transfer_data::= structure
{
        key_id:       enum:
                              (0) global unicast encryption key,
                              (1) global broadcast encryption key,
                              (2) authentication key,
                              (3) master key (KEK)

        key_wrapped:      octet-string
}
```

NOTE 1 It is possible to transfer multiple keys by invoking the method with an array of *n key_transfer data*, or to invoke the method *n* times with an array of a single *key_transfer_data*.

The key wrap algorithm is as specified by the security suite. In suites 0, 1, and 2 the AES key wrap algorithm is used.

The KEK is the master key.

If the unwrapping of the key(s) is successful, the result of the method invocation is success.

If the unwrapping of the key(s) is not successful, the method invocation fails and an appropriate reason for the failure shall be sent back. The keys are not changed.

The new keys are activated immediately after result of the method invocation is sent back with result = success. Notice that this rule equally applies to all keys, including the master key.

NOTE 2 The APDUs carrying the method invocation service primitives are protected as stipulated by the *security_policy* and the access rights to the methods. The method invocation parameters can be additionally protected by invoking the method through a “Data protection” object. The required protection can be specified in project specific companion specifications.

This note equally applies to the *key_agreement (data)* method.

NOTE 3 If using the new keys fails, the client may try to recover from this situation by reverting to using the previous keys or repeating the key transfer process.

4.4.7.3.3 key_agreement (data)

Used to agree on one or more symmetric keys using the key agreement algorithm as specified by the security suite. In the case of suites 1 and 2 the ECDH key agreement algorithm is used with the Ephemeral Unified Model C(2e, 0s, ECC CDH) scheme.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The *data* parameter includes the identification of the key(s) and data used in the key agreement transaction.

```
data::=      array key_agreement_data

key_agreement_data::= structure
{
      key_id:      enum:
            (0) global unicast encryption key,
            (1) global broadcast encryption key,
            (2) authentication key,
            (3) master key (KEK)
      key_data:   octet-string
}
```

NOTE 1 It is possible to agree on multiple keys by invoking the method with an array of *n key_agreement_data*, or to invoke the method *n* times with an array of a single *key_agreement_data*.

The element key_id identifies the key(s) to be agreed on.

The element *key_data* is the public key of ephemeral key pair right concatenated with digital signature, which is calculated over the key_id value and public key of ephemeral key pair value using the digital signature private key: $Q_{e,U}$, II ECDSA(key_id II $Q_{e,U}$).

If the server could verify the digital signature of *key_data*, compute the shared secret and derive the key, then the method invocation is successful and the server sends its own *key_data* to the client. Otherwise, the method invocation fails and a reason for the failure is sent back.

The new keys are activated immediately after result of the method invocation is sent back with result = success.

NOTE 2 If using the new keys fails, the client may try to recover from this situation by reverting to using the previous keys or repeating the key transfer process.

4.4.7.3.4 generate_key_pair (data)

Generates an asymmetric key pair as required by the security suite. The data parameter identifies the usage of the key pair to be generated.

```
data  ::= enum:

(0)  digital signature key pair,
(1)  key agreement key pair,
(2)  TLS key pair
```

NOTE 1 There is maximum one key pair for digital signature, one key pair for key agreement and one key pair for TLS.

NOTE 2 Key agreement key pair is the static key used with the One-Pass Diffie-Hellman C(1e, 1s, ECC CDH) scheme when the server takes the role of Party V or with the Static Unified Model C(0e, 2s, ECC CDH) scheme.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.7.3.5 generate_certificate_request (data)

When this method is invoked, the server sends the Certificate Signing Request (CSR) data that is necessary for a CA to generate a certificate for a server public key. The data parameter identifies the key pair for which the certificate will be requested.

```
data  ::= enum:

(0)  digital signature key pair,
(1)  key agreement key pair,
(2)  TLS key pair
```

NOTE 1 There is maximum one key pair for digital signature, one key pair for key agreement and one key pair for TLS.

The method invocation response parameters include the data necessary to generate the certificate.

NOTE 2 It is the responsibility of the client to forward it to the Certification Authority.

```
response data::= octet-string,
```

The octet-string contains the DER encoding of the CSR as specified in PKCS #10 (RFC 2986).

The CSR shall be signed by the private key belonging to the public key in the request. This acts as the “proof of possession“ of the private key.

4.4.7.3.6 import_certificate (data)

Imports an X.509 v3 certificate of a public key.

```
data  ::= octet-string
```

The octet-string contains the DER encoding of the CSR as specified in PKCS #10 (RFC 2986).

4.4.7.3.7 export_certificate (data)

Exports an X.509 v3 certificate in the server.

Certificate is identified with entity identification or the serial number of the certificate.

```
data::= certificate_identification

certificate_identification::= structure
{
    certificate_identification_type:      enum:
                                  (0) certificate_identification_entity,
                                  (1) certificate_identification_serial

    certification_identification_options: CHOICE
    {
        certificate_identification_by_entity,
        certificate_identification_by_serial
    }
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
certificate_identification_by_entity::= structure
{
        certificate_entity: enum:
                                    (0) server,
                                    (1) client,
                                    (2) certification authority,
                                    (3) other
        certificate_type: enum:
                                    (0) digital signature,
                                    (1) key agreement,
                                    (2) TL
                                    (3) other

        system_title:               octet-string
}

certificate_identification_by_serial::= structure
{
        serial_number:              octet-string,
        issuer:                     octet-string
}

response data::= octet-string
response data octet-string is formatted as X.509 v3 DER format.
```

If no matching certificate is found, the response shall contain an appropriate error code.

4.4.7.3.8 remove_certificate (data)

Removes X.509 v3 certificate in the server.

```
data::= certificate_identification

certificate_identification see 4.4.7.3.7.
```

NOTE The certificates of the server for digital signature, key agreement and TLS cannot be removed from the server. When update of these certificates is needed the new key pair is generated, new certificate signing request is generated and new certificate is imported.

4.4.8 Push interface classes and objects

4.4.8.1 Overview

There are several occasions on which DLMS® messages can be ‘pushed’ to a destination without being explicitly requested. The push is initiated by a trigger such as.:

- if a scheduled time is reached;
- if a value being locally monitored exceeds a threshold;
- a local event (e.g. power-up/down, push button pressed, meter cover opened).

The DLMS®/COSEM push mechanism follows the publish/subscribe pattern:

Publishing is modelled in DLMS®/COSEM by the *object_list* attribute of “Association” objects which provides the list of COSEM objects and their attributes accessible in a given AA. Subscription is modelled by writing the appropriate attributes of “Push setup” objects. The required data are sent – upon the trigger specified – using the xDLMS DataNotification service when the specified trigger is activated.

The COSEM model of the Push operation is shown in Figure 12.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

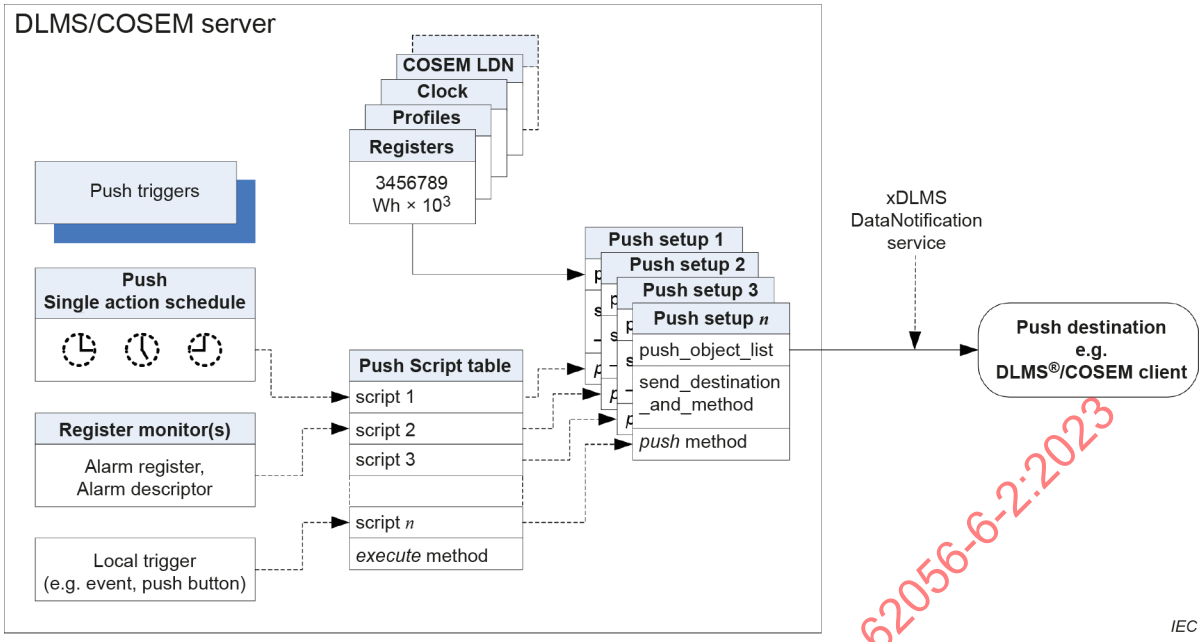


Figure 12 – COSEM model of push operation

The core element of modelling the push operation is the "Push setup" IC. The *push_object_list* attribute contains a list of references to COSEM object attributes to be pushed.

When push uses a gateway, then the version of the gateway protocol for end devices without WAN/NN knowledge is to be applied. This is described in the IEC 62056-5-3:2023, 10.9.4.3.

The various triggers (e.g. schedulers, monitors, local triggers etc.) call a script entry in a Push "Script table" object (new instances of the "Script table" IC) which invokes then the *push* method of the related "Push setup" object. The destination, the communication media, the protocol, the encoding, the timing as well as any retries of the push operation are determined by the other attributes of the "Push setup" object.

Each trigger can cause the data to be sent to a dedicated destination. Therefore, for every trigger or a group of triggers an individual "Push setup" object is available defining the content and the destination of the push message as well as the communication medium used.

For the purposes of pushing data when an alarm occurs, Alarm monitor objects (new instances of the "Register monitor" IC) are available.

The alarms are held by *Alarm register* or *Alarm descriptor* objects, see 6.2.64.

The structure of the *Alarm descriptor* as well as the alarm conditions needs to be defined in a project specific companion specification.

Alarm descriptor objects are monitored by Alarm "Register monitor" objects, see 6.2.13. The *actions* attribute provides the link to the action_up and maybe the action_down scripts in the Push "Script table" object – see 6.2.7 – which invoke then the *push* method of the desired "Push setup" object. When an alarm occurs, the predefined set of data – that may include among other data the *Alarm register* and *Alarm descriptor* objects – are pushed.

The push data is sent – when the conditions for pushing are met – using the unsolicited, non-client/server type xDLMS service, the DataNotification service. When the data pushed is long, it can be sent in blocks.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The push process takes place within the application context of the AA in which the “Push setup” object is visible. The security context is determined by the “Security setup” object referenced from the “Association” object. The necessary data protection parameters are defined in attribute *push_protection_parameters* which offers the same options as defined in the “Data Protection” IC.

NOTE Details are subject to project specific companion specifications.

All information necessary to process the data received by the client shall be either:

- retrieved by the client e.g. by reading the *push_object_list attribute*; or
- shall be part of the data pushed; or
- shall be predefined in the client application.

4.4.8.2 Push setup (class_id = 40, version = 2)

4.4.8.2.1 Overview

The Push setup IC contains a list of references to COSEM object attributes to be pushed. It also contains the push destination and method as well as the communication time windows and the handling of retries.

In version 1 the possibility of data protection was added offering the same options as defined in the Data protection IC.

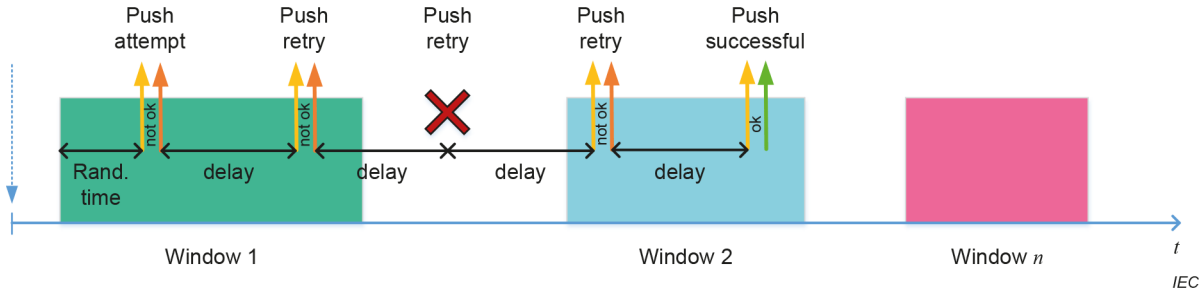
In version 2 six new possibilities are specified:

- a new entry selection mechanism allows selecting entries related to the last confirmed entry;
- a new column selection mechanism allows explicitly selecting columns of Profile generic object buffer attributes;
- the **repetition_delay** attribute now provides an exponential formula that allows increasing the repetition delay with each retry attempt;
- a new **push_operation_method** attribute allows specifying if the DataNotification.request service primitive is invoked with Service_Class == Unconfirmed or confirmed and how the push retry operation works;
- a new **confirmation_parameters** attribute allows limiting the time interval in which entries related to last confirmed entry can be selected;
- a new **last_confirmation_date_time** attribute holds the date_time when the DataNotification service was last confirmed.

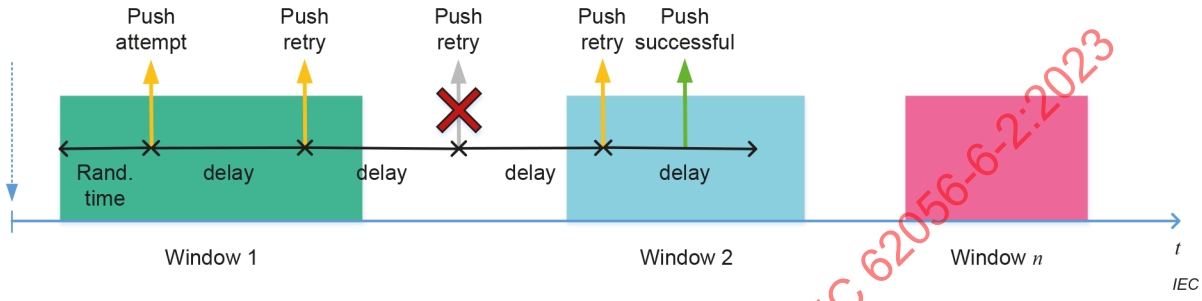
This version of the interface class is intended to facilitate the use of relative and absolute data selection. It would be common practice to use an instance of the Push setup IC with relative data selection for routine data push, and an instance of the Push setup IC with absolute data for special cases.

The push is started when the *push* method is invoked, triggered by a Push “Single action schedule” object, by an alarm “Register monitor” object, by a dedicated internal event or externally. After the push operation has been triggered, it is executed according to the settings made in the given “Push setup” object. Depending on the communication window settings, the push is executed immediately or as soon as a communication window becomes active, after a random delay. If the push was not successful, retries are made. Push retries may be made when supporting protocol layer failure is indicated or when confirmation is missing. Push operation with windows, delays and retries is shown in Figure 13.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



Case a) Retry on supporting layer failure



Case b) Retry on missing confirmation

Figure 13 – Push windows and delays

Push setup		0...n	class_id = 40, version = 2			
Attribute (s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. push_object_list	(static)	array				x + 0x08
3. send_destination_and_method	(static)	structure				x + 0x10
4. communication_window	(static)	array				x + 0x18
5. randomisation_start_interval	(static)	long-unsigned				x + 0x20
6. number_of_retries	(static)	unsigned				x + 0x28
7. repetition_delay	(static)	structure				x + 0x30
8. port_reference	(static)	octet-string				x + 0x38
9. push_client_SAP	(static)	integer				x + 0x40
10. push_protection_parameters	(static)	array				x + 0x48
11. push_operation_method	(static)	enum				x + 0x50
12. confirmation_parameters	(static)	structure				x + 0x58
13. last_confirmation_date_time	(dyn.)	date-time				x + 0x60
Specific methods		***m/o***				
1. push (data)		m				x + 0x38
2. reset (data)		o				x + 0x70

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8.2.2 Attribute description

4.4.8.2.2.1 logical_name

Identifies the “Push setup” object instance. See 6.2.24.

4.4.8.2.2.2 push_object_list

Defines the list of attributes to be pushed.

Upon invocation of the push (data) method the items are sent to the destination defined in the send_destination_and_method attribute.

```
array        push_object_definition

push_object_definition::=      structure
{
             class_id:         long-unsigned,
             logical_name:     octet-string,
             attribute_index:  integer,
             data_index:       long-unsigned,
             restriction:      restriction_element,
             columns:          column_element
}

      restriction_element::= structure
      {
             restriction_type:  enum:
                          (0) none,
                          (1) restriction by date,
                          (2) restriction by entry

             restriction_value: CHOICE
             {
             null-data, // no restrictions apply
             restriction_by_date,
             restriction_by_entry
             }
      }
      restriction_by_date::= structure
      {
             from_date:  octet-string,
             to_date:    octet-string
      }
      restriction_by_entry::= structure
      {
             from_entry: double-long-unsigned,
             to_entry:   double-long-unsigned
      }
      column_element::= array capture_object_definition
```

NOTE 1 For capture_object_definition, see 4.3.6.2.3.

Where:

– class_id, logical_name and attribute_index reference a COSEM object attribute. For referencing all attributes of an object, see IEC 62056-5-3:2023, 9.1.4.3.7, attribute_0 referencing.

If the attribute is simple or if attribute_0 is referenced, then:

– data_index has no meaning and shall be set to 0x0000;

– restriction_element restriction_type shall be 0 (none) and restriction_element restriction_value shall be null-data;

– column_element shall be an array of 0 elements.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If the attribute is structure or an array – other than Profile generic object buffer then:

– data_index points to one element in the structure or array and can take any value between 0x0001 and 0x0FFF;

– restriction_element restriction_type shall be 0 (none) and restriction_element restriction_value shall be null-data;

– column_element shall be an array of 0 elements.

If the attribute is a Profile generic object buffer, then data_index, restriction and column define selective access parameters to select entries and columns.

There are three, mutually exclusive entry selection mechanisms available:

1) Relative selective access related to current date and time. Entries in last time intervals relative to current date and time or last entries as identified by data_index are selected;

2) Relative selective access related to last confirmed entry. Entries in next time intervals relative to last confirmed entry or next entries as identified by data_index are selected;

3) Absolute selective access. Entries in an explicitly defined date range or entry range as identified by restriction are selected.

There are two, mutually exclusive column selection mechanisms available:

a) a defined number of columns starting from column 1 are selected;

b) an explicitly defined list of columns are selected.

In the case of entry selection mechanism (1) and (2) the selective access parameters are defined by data_index interpreted as three different fields as shown below. The encoding is shown in Table 21.

data_index	MS-Byte		LS-Byte
	Upper nibble	Lower nibble	

Where:

– data_index MS-Byte Upper nibble defines how the entries are selected;

– data_index MS-Byte Lower nibble is used to specify the number of columns when applicable;

– data_index LS-Byte defines the number of entries or time intervals.

In the case of entry selection mechanism (1) and (2), restriction_element restriction_type shall be 0 and restriction_element restriction_value shall be null_data.

In the case of entry selection mechanism (3) the selective access parameters are defined by restriction_element. In this case Data_index MS_byte Upper_nibble shall be set to 0x0, and Data_index LS_byte shall be set to 0x00.

– for restriction by date range restriction_type holds (1) range by date and restriction_value holds restriction_by_date structure;

– for restriction by entries restriction_type holds (2) range by entry and restriction_value holds restriction_by_entry structure;

  if selective access is not required, restriction_type holds (0) none and restriction_value holds null-data. This choice shall be taken also if relative selective access is used.

In the case of column selection mechanism (a), Data_index MS_byte Lower_nibble defines the number of columns to be selected, starting from column 1. In this case, column_element shall be an empty array.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

In the case of column selection mechanism (b), the columns to be selected are identified by column_element. In this case, Data_index MS_byte Lower_nibble shall be set to 0.

If relative selective access related to last confirmed entry is used, the last confirmed entry is updated when the DataNotification.confim service primitive is invoked with Result == CONFIRMED.

The DataNotification APDU carrying the push data is protected as stipulated by the security suite, security policy and the security material held by the "Security setup" object referenced from the Association SN / Association LN object within the application context of the AA which is linked to the push_client_SAP attribute; as well as by the push_potection_parameters attribute. For attributes to which no access right is granted within this AA, or which cannot be accessed for any other reason, null-data should be returned.

NOTE 2 If the push_object_list array is empty, the push operation is disabled.

NOTE 3 The push_object_list attribute itself can be also pushed to identify the data pushed.

NOTE 4 Last confirmed entry is an internal variable maintained by the server AP.

4.4.8.2.2.3 send_destination_and_method

Contains the destination address (e.g. phone number, email address, IP address) where the data specified by the *push_object_list* has to be sent, as well as the sending method.

send_destination_and_method::= structure

```
{
    transport_service:    transport_service_type,
    destination:          octet-string,
    message:              message_type
}
```

Where:

– the transport_service element defines the type of service used to push the data:

transport_service_type::= enum:

```
(0)         TCP,
(1)         UDP,
(2)         reserved for FTP,
(3)         reserved for SMTP,
(4)         SMS,
(5)         HDLC,
(6)         reserved for M-Bus,
(7)         reserved for ZigBee®
(200...255) manufacturer specific
```

– the destination element contains the target address where the data has to be sent. The elements of the target address depend on the transport service used.

Each "Push setup" object instance specifies a single destination. If it is required to push data to several destinations, several "Push setup" objects have to be instantiated,

– the message_type element identifies the encoding of the xDLMS APDU used.

message_type::= enum:

```
(0)         A-XDR encoded xDLMS APDU,
(1)         XML encoded xDLMS APDU,
(128...255) manufacturer specific
```

– All other transport_service_type and message_type values are reserved for future use.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8.2.2.4 communication_window

Defines the time points when the communication window(s) for the push become(s) active (start_time) and inactive (end_time). See Figure 13.

```
array        window_element

window_element::= structure
{
        start_time:  octet-string,
        end_time:    octet-string
}
```

start_time and end_time are formatted as specified in 4.1.6.1 for *date-time* including wildcards.

If the end of a communication window is reached when a push has already started the operation will be completed.

If no communication windows are defined (array [0]) the push operation is always possible.

4.4.8.2.2.5 randomisation_start_interval

Defines a maximum delay, in seconds, of sending the first DataNotification APDU after invoking the push method. This is to avoid a situation where many servers may push simultaneously. It is used as follows:

- the delay is random from 0 up to the maximum value. A value of 0 deactivates the mechanism;
- if the push is invoked within an active communication window, the random delay is applied. It is not applied for retries;
- if the push is invoked outside a communication window, then the random delay is applied when the communication windows opens;
- if the random delay extends beyond the current communication window, then the DataNotification APDU will be sent in the next window if available;
- if no communication windows are defined, then the random delay is applied whenever the push method is invoked including all retries.

4.4.8.2.2.6 number_of_retries

Defines the maximum number of retries in the case of unsuccessful or skipped push attempts. After a successful push operation no further push attempts are made until the push operation is triggered again. For further details on the retransmission operation see the *push_operation_method* attribute (4.4.8.2.2.11).

4.4.8.2.2.7 repetition_delay

Defines the elements for calculating the time delay until the next attempt after an unsuccessful push. See Figure 13.

```
repetition_delay::= structure
{
        repetition_delay_min:          long-unsigned,
        repetition_delay_exponent:     long-unsigned,
        repetition_delay_max           long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

- repetition_delay_min is the minimum delay, in seconds, until a next push attempt is started;
- repetition_delay_exponent is used for calculating the next delay;
- repetition_delay_max is the maximum delay, in seconds. Any calculated time delay will be capped by this value.

The repetition delay is calculated using the following formula:

$$repetition_delay = repetition_delay_min \times (repetition_delay_exponent \times 0.01)^{n-1}$$

where n refers to the ordinal number of the push retries with n = 1 for the first retry etc.

NOTE 1 The repetition delay itself is not influenced by the communication window. But a push retry only can be made if a communication window is active at that time. Otherwise it is handled like an unsuccessful push attempt.

NOTE 2 The push data is not stored in an intermediate buffer. In the case of push retries, the current values of the attributes may change with every push retry attempt.

4.4.8.2.2.8 port_reference

Contains the logical name of a communication port setup object allowing the selection of a specific communication channel for the push based on the transport_service_type. This mainly applies in cases where several channels of the same type are supported.

If this information is not available or not needed, the attribute may be left empty (octet-string [0]).

4.4.8.2.2.9 push_client_SAP

Defines the client SAP where the push is directed to in the supporting layer of the DataNotification service. The push process takes place within the application context of the AA which is linked to the push_client_SAP attribute. The security context is determined by the Security setup object referenced in the related Association SN /LN object.

4.4.8.2.2.10 push_protection_parameters

Specifies all protection parameters to be applied to the DataNotification APDU when the push data is sent. It offers the same options as the Data Protection IC but it is bound to the sending of the data defined in the push_object_list attribute.

```
array protection_parameters_element

        protection_parameters_element::= structure
        {
            protection_type: enum:
                        (0) authentication,
                        (1) encryption,
                        (2) authentication and encryption,
                        (3) digital signature

            protection_options: structure
            {
                transaction_id:             octet-string,
                originator_system_title:    octet-string,
                recipient_system_title:     octet-string,
                other_information:          octet-string,

                key_info:   key_info_element
            }
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
}
```

Where:

– transaction_id holds the identifier of the transaction;

– originator_system_title holds the system title of the originator that applies the protection;

– recipient_system_title holds the system title of the recipient which will check and remove the given protection;

– other_information carries other information. Its contents may be specified in project specific companion specifications. An octet-string of length 0 indicates that this field is not used;

– key_info holds the information necessary for the recipient to obtain the right key for checking and removing authentication and encryption. In the case of digital signature, key_info is not necessary and it shall be a structure of 0 elements.

The fields transaction-id….other-information are A-XDR encoded OCTET STRINGs. The length and the value of each field are included in the AAD when applicable.

```
key_info_element::= structure
{
        key_info_type: enum:
                        (0) identified_key,
                        -- used with identified_key_info_options

                        (1) wrapped_key,
                        -- used with wrapped_key_info_options

                        (2) agreed_key
                        -- used with agreed_key_info_options

        key_info_options: CHOICE
        {
                identified_key_info_options,
                wrapped_key_info_options,
                agreed_key_info_options
        }
}
identified_key_info_options::= enum:
                        (0) global_unicast_encryption_key,
                        (1) global_broadcast_encryption_key

wrapped_key_info_options::= structure
{
        kek_id:         enum:
                        (0) master_key
        key_ciphered_data: octet-string
}
agreed_key_info_options::= structure
{
        key_parameters: octet-string,
        key_ciphered_data: octet-string
}
```

This attribute is first written by the client. The server may need to fill in some additional elements.

The use of the various elements is the same as specified in Table 22 and Table 23 of the Data protection IC (class_id = 30, version = 0).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8.2.2.11 push_operation_method

Defines if the DataNotification.request service primitive is invoked with Service_Class == Unconfirmed or Confirmed and how the push retry operates.

enum:

(0) unconfirmed, retry on supporting protocol layer failure,
(1) unconfirmed, retry on missing supporting protocol layer confirmation,
(2) confirmed, retry on missing confirmation.

In case (0), the repetition delay for the next retry attempt is started upon supporting layer failure reported through the DataNotification.confirm service primitive.

In cases (1) and (2), the repetition delay for the next retry attempt is started when the DataNotification.request service primitive is invoked.

In cases (0) and (1), the push operation is deemed as successful when supporting layer confirmation is reported by invoking DataNotification.confim service primitive with Result == CONFIRMED.

In case (2), the push operation is deemed as successful when the server AL receives the data-notification-confirm APDU and invokes DataNotification.confirm service primitive with Service_Class == Confirmed and Result == CONFIRMED.

4.4.8.2.2.12 confirmation_parameters

Defines the selection of entries defined by data_index to avoid pushing data from too far in the past. If entries have not yet been confirmed then all entries are selected. This attribute is applicable only for relative selective access related to last confirmation.

```
confirmation_parameters::= structure
{
    confirmation_start_date:    date-time,
    confirmation_interval:      double-long-unsigned
}
```

Where:

- confirmation_start_date is the starting date and time for selecting entries. Fields of date-time not specified are not used. If all fields are not specified the use of confirmation_start_date is disabled;
- confirmation_interval is a time interval in seconds backwards from the current date and time. If confirmation_interval is set to zero, the use of confirmation_interval is disabled.

4.4.8.2.2.13 last_confirmation_date_time

Holds the date and time when the AL most recently invoked the DataNotification.confirm service primitive with Result == CONFIRMED.

date-time is formatted as specified in 4.1.6.1.

4.4.8.2.3 Method description

4.4.8.2.3.1 push (data)

Activates the push process leading to an attempt to send a DataNotification APDU carrying the push data.

```
data::= integer(0)
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8.2.3.2 reset (data)

Resets the push process to initial state.

data::= integer(0)

4.4.8.2.4 Encoding of selective access parameters with data_index

Table 21 provides details of how selective access parameters are defined using the data_index.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 21 – Encoding of selective access parameters with data_index

	MS_Byte upper nibble: Selects the time intervals or entries.
0xF	**Last complete number of months:** Selective access to the *buffer* resulting in all entries of the last complete number of months and the first entry at midnight of the current month.
0xE	**Last complete number of days:** Selective access to the *buffer* resulting in all entries of the last complete number of days and the first entry at midnight of today.
0xD	**Last complete number of hours:** Selective access to the *buffer* resulting in all entries of the last complete number of hours and the first entry of the current hour.
0xC	**Last complete number of minutes:** Selective access to the *buffer* resulting in all entries of the last complete number of minutes and the first entry of the current minute.
0xB	**Last number of seconds:** Selective access to the *buffer* resulting in all entries of the last number of seconds.
0xA	**Last complete number of months including the current month:** Same as 0xF above but all entries up to now are retrieved.
0x9	**Last complete number of days including the current day:** Same as 0xE above but all entries up to now are retrieved.
0x8	**Last complete number of hours including the current hour:** Same as 0xD above but all entries up to now are retrieved.
0x7	**Last complete number of minutes including the current minute:** This is the same as 0xC above but all entries up to now are retrieved.
0x6…0x2	Values 0x6 to 0x2 apply only for the Push interface class version 2 (see 4.4.8.2).
0x6	**Complete number of months after last confirmation:** This is equal to a selective access to the profile *buffer* resulting in all entries of the complete number of months after the last confirmed entry.
0x5	**Complete number of days after last confirmation:** This is equal to a selective access to the profile *buffer* resulting in all entries of the complete number of days after the last confirmed entry.
0x4	**Complete number of hours after last confirmation:** This is equal to a selective access to the profile *buffer* resulting in all entries of the complete number of hours after the last confirmed entry.
0x3	**Complete number of minutes after last confirmation:** This is equal to a selective access to the profile *buffer* resulting in all entries of the complete number of minutes after the last confirmed entry.
0x2	**Number of entries after last confirmation**: This is equal to a selective access to the profile *buffer* resulting in number of entries after the last confirmed entry.
0x1	**Last number of entries**
0x0	**Whole attribute** or a single element in an attribute with complex data type is selected (MS-Byte lower nibble 0x0…0xF, LS-Byte 0x00…0xFF).
	MS-Byte lower nibble !=0: Defines the number of columns of a “Profile generic” *buffer* selected.
0x0	All columns
0x1 to 0xF	Number of columns, starting from column 1.
0x00…0xFF	LS-Byte: – when entries in time intervals are selected, specifies the number of time intervals, value 0x00 defines all time periods; – when entries are selected, specifies the number of entries, value 0x00 defines all entries.
Example 1)	0xE401: The entries for the last complete day are selected. The first 4 columns are included.
Example 2)	0xA300: The entries for zero last complete months and the current month are selected . The first 3 columns are included.
Example 3)	0x800C: The entries for the last complete 12 hours are selected. All columns are included.
Example 4)	0x1080: The last 128 entries are selected. All columns are included.
Example 5)	0x4608: The entries for the last complete 8 hours after the last confirmed entry are selected. The first 6 columns are included.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9 COSEM data protection (class_id = 30, version = 0)

4.4.9.1 Overview

Instances of this IC allow applying cryptographic protection on COSEM data i.e. on attribute values and method invocation and return parameters. This is achieved by accessing attributes and/or methods of other COSEM objects indirectly through instances of the “Data protection” interface class that provide the necessary mechanisms and parameters to apply / verify / remove protection on COSEM data.

NOTE 1 “Accessing” includes reading / writing / capturing / pushing COSEM object attributes or invoking methods.

NOTE 2 When attributes and methods of COSEM objects are accessed directly, protection can be provided by protecting the xDLMS APDUs as stipulated by the relevant security policy and the access rights.

NOTE 3 For definitions and abbreviations related to cryptographic security see IEC 62056-5-3:2023, Clause 3.

Protection on COSEM data is aligned with and complements protection on xDLMS APDUs as defined in IEC 62056-5-3:2023, 5.7.

The use cases for COSEM data protection include, but are not limited to:

- Reading or writing a pre-defined set of protected attribute values;
- storing a pre-defined set of protected attribute values in “Profile generic” objects for later retrieval;
- pushing a pre-defined set of protected attribute values;
- reading or writing selected attributes of other COSEM objects with protection;
- invoking a method of another COSEM object with protected method invocation and return parameters.

Protection may comprise any combination of authentication, encryption and digital signature and can be applied in a layered manner. The parties applying and removing the protection are the DLMS®/COSEM server and another identified party, which may be a DLMS®/COSEM client or a third party.

Applying data protection between a DLMS®/COSEM server and a third party allows keeping critical / sensitive data confidential towards the client through which the third party accesses the server. Signing COSEM data by a third party supports non-repudiation.

For end-to-end protection between third parties and servers, see also IEC 62056-5-3:2023, 4.1.7 and 5.2.5.

The protection parameters are always controlled by the client with some elements filled in by the server as appropriate.

The security suite is determined by the “Security setup” object referenced from the current “Association SN” / “Association LN” object.

Figure 14 shows the COSEM model of data protection and the relationship of a “Data protection” object with other COSEM objects.

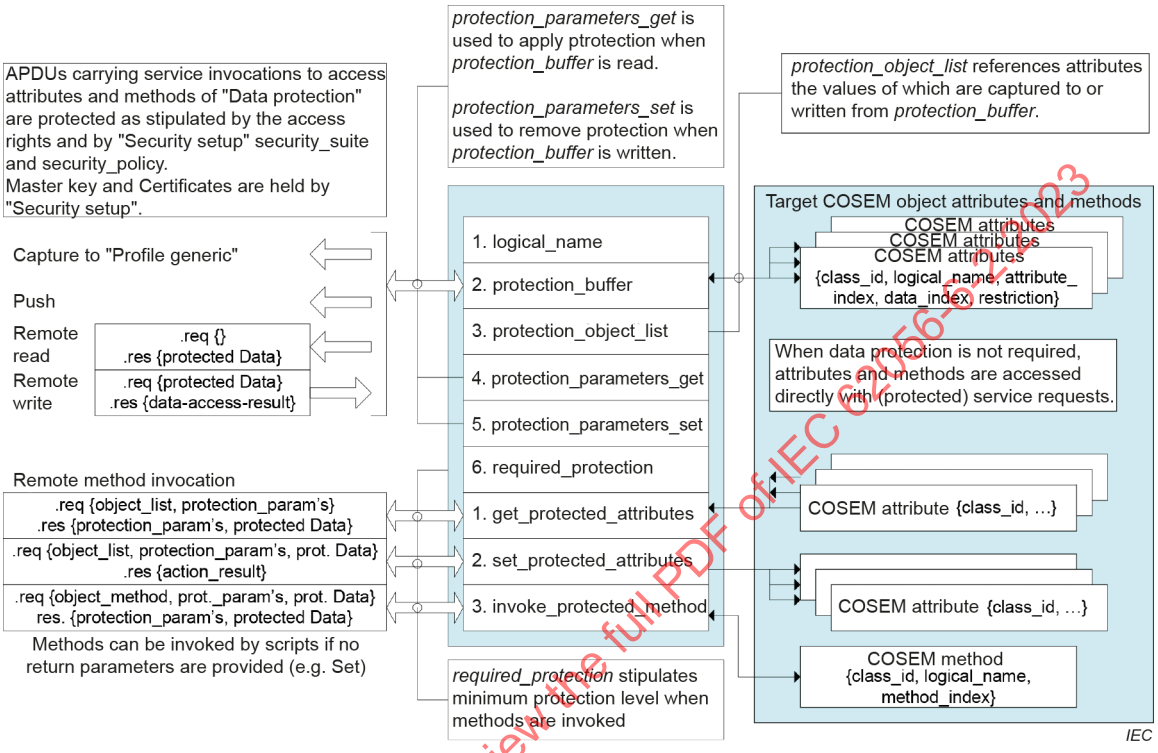
For accessing attributes of other COSEM objects with protected data, there are two mechanisms available:

- reading or writing the *protection_buffer* attribute. The *protection_buffer* can be also captured in “Profile generic” objects or pushed using “Push setup” objects;
- invoking the *get_protected_attributes* / *set_protected_attributes* method.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

For accessing a method of another COSEM object with protected data, the *invoke_protected_method* method is available.

APDUs carrying service invocations to access attributes and methods of “Data protection” objects are protected as stipulated by access rights to these attributes and methods, and by “Security setup” *security_suite* and *security_policy*.



The master key and Certificates – as required by the security suite – are held by “Security setup”.

Figure 14 – COSEM model of data protection

Protection on COSEM data is applied and removed in the various cases as follows:

a) When the *protection_buffer* attribute is read / captured in a “Profile generic” object / pushed:
   - attributes determined by *protection_object_list* are captured;
   - protection according to *protection_parameters_get* is applied on the set of attributes and the result is put to *protection_buffer*;
   - the value of *protection_buffer* is returned / captured in the “Profile generic” *buffer* / pushed using “Push setup” objects.

b) When the *protection_buffer* is written:
   - protected Data are written to *protection_buffer;* and
   - protection according to *protection_parameters_set* is removed and the resulting attribute values are written to the attributes specified by *protection_object_list.*

c) When the *get*_protected_*attributes* method is invoked:
   - attributes determined by the object_list element of get_protected_attributes_request are captured;
   - protection according to the *required_protection* attribute and response protection_parameters is applied. If protection_parameters do not satisfy required_protection then the method invocation fails;
   - the protected attribute values are returned.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

d) When the set_*protected_attributes* method is invoked:

- protection on protected_attributes is verified and removed using the protection_parameters that shall meet *required_protection*;
- the resulting attribute values are put in the attributes specified by the object_list element;

e) When the *invoke_protected_method* method is invoked:

- protection from protected method invocation parameters is removed using the protection parameters in the request that shall meet *required_protection*;
- the method specified by the object_method element of invoke_protected_method_request is invoked with this method invocation parameter;
- on the return parameters, the protection using the response protection parameters that shall meet *required_protection* is applied. If protection_parameters do not satisfy required_protection then the method invocation fails;
- the protected method return parameters are returned.

Figure 15 shows, as an example, how protected Data in *protection_buffer* is constructed from the attributes determined by *protection_object_list* according to the *protection_parameters_get.* See also IEC 62056-5-3:2023, Figure 29.

When the *protection_buffer* attribute is read the following steps are performed:

f) prerequisites: *protection_object_list*, *protection_parameters_get*, master key, key agreement and digital signature certificates as needed;

g) capture COSEM object attributes determined by *protection_object_list* and create Data, a structure containing the individual Data of the attributes captured;

h) protect Data according to *protection_parameters_get.*

NOTE 4 In the example shown in Figure 15 two layers of protection are applied:

– the first layer is a combination of compression / encryption / authentication as determined by the Security control byte SC, resulting (C)Data,

– the second layer is digital signature applied to (C)Data.

i) put the protected data, of data type octet-string, into *protection_buffer*;

j) return the value of *protection_buffer*.

It may be necessary to read also protection_parameters_get to obtain the protection parameters to verify / remove protection by the recipient.

The invocation counter used when protection is applied / removed is related to the key used. When the protection is applied the corresponding invocation counter is incremented. When the key is changed the invocation counter shall be reset to 0.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

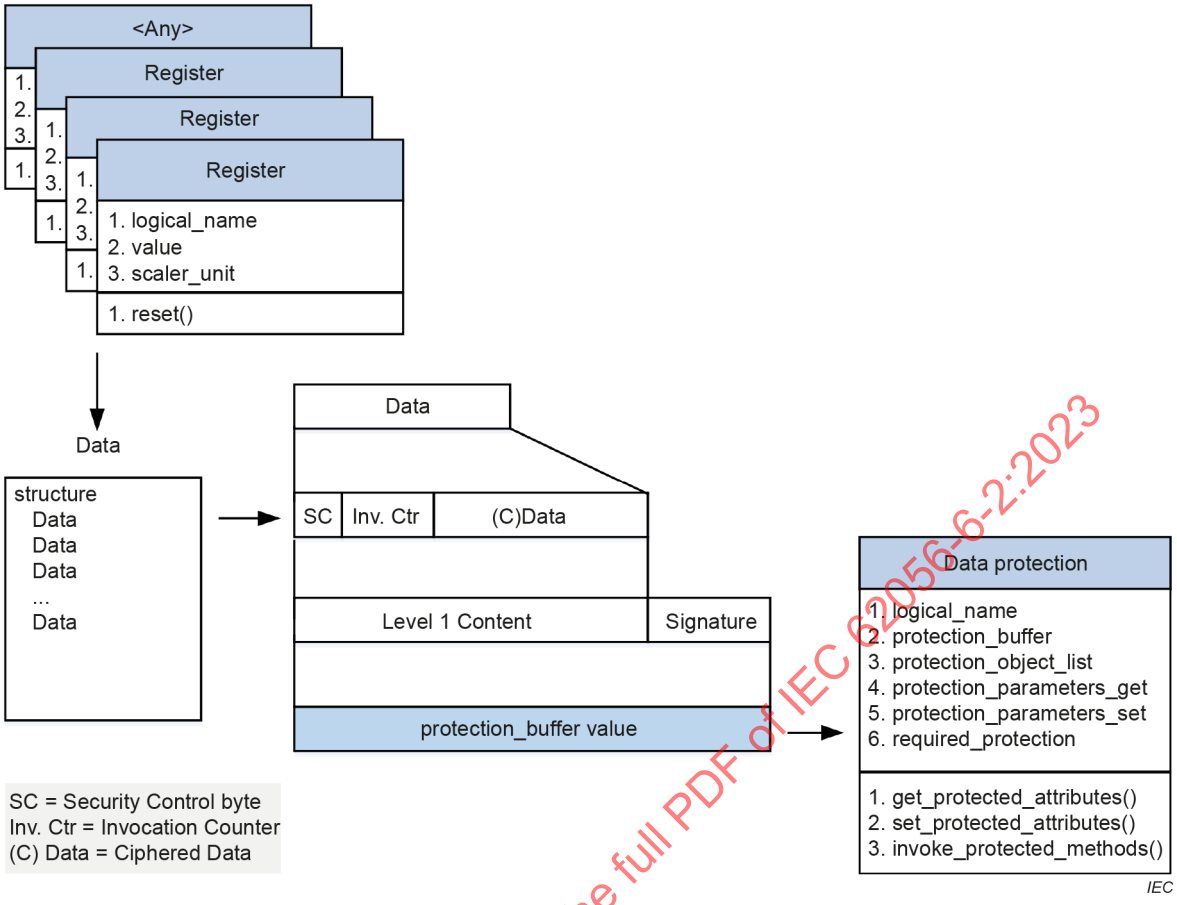


Figure 15 – Example: Read *protection_buffer* attribute

Data protection		**0...n**	**class_id = 30, version = 0**			
Attribute (s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. protection_buffer	(dyn.)	octet-string				x + 0x08
3. protection_object_list	(static)	array				x + 0x10
4. protection_parameters_get	(static)	array				x + 0x18
5. protection_parameters_set	(static)	array				x + 0x20
6. required_protection	(static)	enum				x + 0x28
Specific methods		***m/o***				
1. get_protected_attributes (data)		m				x + 0x30
2. set_protected_attributes (data)		m				x + 0x38
3. invoke_protected_method (data)		m				x + 0x40

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9.2 Attribute description

4.4.9.2.1 logical_name

Identifies the “Data protection” object instance. See 6.2.36.

4.4.9.2.2 protection_buffer

Contains the protected Data.

When read, the attributes determined by *protection_object_list* are captured then protection is applied according to *protection_parameters_get*.

When written, the protected Data is put into the *protection_buffer*, then the protection is verified / removed according to *protection_parameters_set* and the attributes determined by *protection_object_list* are set.

4.4.9.2.3 protection_object_list

Defines the list of attributes to be captured to *protection_buffer* when it is read or the list of attributes to be set when *protection_buffer* is written.

Two, mutually exclusive selective access mechanisms are available:

– relative selective access, i.e. entries defined relative to current date or entry are returned: this mechanism is controlled by the data_index element; or

– absolute selective access, i.e. entries in an explicitly defined date range or entry range are returned: this mechanism is controlled by the restriction element.

```
array object_definition

    object_definition::=    structure
    {
        class_id:               long-unsigned,
        logical_name:           octet-string,
        attribute_index:        integer,
        data_index:             long-unsigned,
        restriction:            restriction_element
    }
```

Where:

– attribute_index is a pointer to the attribute within the object, identified byy class_id and logical_name: attribute_index 1 refers to the 1st attribute (i.e. the *logical_name*), attribute_index 2 to the 2nd attribute etc.; attribute_index 0 refers to all public attributes;

– data_index is a pointer selecting one or several specific elements of an attribute with a complex data type (structure or array):

- if the data type of the attribute is simple, then data_index has no meaning;
- if the data type of the attribute is a structure or an array, then data_index points to one or several specific elements in the structure or array;
- when the attribute is the *buffer* of a “Profile generic” object, the data_index carries selective access parameters relative to current date or entry.

data_index:	MS-Byte		LS-Byte
	Upper nibble	Lower nibble	

– 0x0000 = identifies the whole attribute;

– 0x0001 to 0x0FFF = identifies one element in the complex attribute. The first element in the complex attribute is identified by data_index 1;

– 0x1000 to 0xFFFF = selective access to the array holding the buffer of a “Profile generic” object. The data-index selects entries within a number of last (recent) time periods, or a number of last (recent) entries, as well as the columns in the array.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The encoding is specified in Table 25.

When the attribute is the buffer of a “Profile generic” object, then restriction_element specifies selective access parameters in an explicitly defined date range or entry range.

```
restriction_element::=   structure
{
        restriction_type:  enum:
                    (0)    none,
                    (1)    restriction by date,
                    (2)    restriction by entry
        restriction_value: CHOICE
        {
              null-data,                  // no restrictions apply
              restriction_by_date,
              restriction_by_entry
        }
}

restriction_by_date::=   structure
{
        from_date:         octet-string,
        to_date:           octet-string
}

restriction_by_entry::=  structure
{
        from_entry:        double-long-unsigned,
        to_entry:          double-long-unsigned
}
```

– restriction_element defines absolute selective access to a “Profile generic” buffer by date range (from_date to to_date) or by entries (from_entry to to_entry). To use this absolute selective access mechanism, data_index shall be:
  - MS Byte upper nibble set to 0x0;
  - MS Byte lower nibble set to 0x0 to 0xF in accordance with Table 21;
  - Lower Byte upper nibble set to 0x00;
– restriction_element is composed of restriction_type and restriction_value:
  - for restriction by date range the restriction_type element holds (1) restriction by date and the restriction_value element holds restriction_by_date structure;
  - for restriction by entries the restriction_type element holds (2) restriction by entry and the restriction_value element holds restriction_by_entry structure;
  - otherwise, the restriction_type element holds (0) none and the restriction_value element holds null-data. This choice shall be taken also if relative selective access is to be used.

4.4.9.2.4 protection_parameters_get

Contains all necessary parameters to specify the protection applied when the *protection_buffer* is read.

```
array protection_parameters_element

protection_parameters_element::= structure

{
        protection_type:  enum:
authentication,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
encryption,
authentication and encryption,
digital signature
        protection_options:  structure
        {
                transaction_id:                 octet-string,
                originator_system_title:        octet-string,
                recipient_system_title:         octet-string,
                other_information:              octet-string,
                key_info:                       key_info_element
        }
}
```

Where:

– transaction_id holds the identifier of the transaction;

– originator_system_title holds the system title of the originator that applies the protection;

– recipient_system_title holds the system title of the recipient which will check and remove the given protection;

– other_information carries other information. Its contents may be specified in project specific companion specifications. An octet-string of length 0 indicates that this field is not used;

– key_info holds the information necessary for the recipient to obtain the right key for checking and removing authentication and encryption. In the case of digital signature, key_info is not necessary and it shall be a structure of 0 elements.

The fields transaction-id ….other-information are A-XDR encoded OCTET STRINGs. The length and the value of each field are included in the AAD when applicable.

```
key_info_element::= structure
{
        key_info_type:    enum:
                        (0) identified_key,
                                -- used with identified_key_info_options
                        (1) wrapped_key,
                                -- used with wrapped_key_info_options
                        (2) agreed_key
                                -- used with agreed_key_info_options
        key_info_options: CHOICE
        {
                identified_key_info_options,
                wrapped_key_info_options,
                agreed_key_info_options
        }
}
identified_key_info_options::=      enum:
global_unicast_encryption_key,
global_broadcast_encryption_key

wrapped_key_info_options::=structure
{
        kek_id:                 enum:
                        (0) master_key,

        key_ciphered_data:      octet-string
}
agreed_key_info_options::=  structure
{
        key_parameters:         octet-string,
        key_ciphered_data:      octet-string
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

This attribute is first written by the client. The server may need to fill in some additional elements, in which case this attribute has to be read back by the client – and if needed, forwarded to the third party – so that they can use these parameters to verify / remove protection.

For the use of the various elements see Table 22 and Table 23.

4.4.9.2.5 protection_parameters_set

Contains all necessary parameters to verify / remove the protection applied when the *protection_buffer* is written.

array protection_parameters_element

protection_parameters_element: see the *protection_parameters_get* attribute (4.4.9.2.4).

This attribute is written by the client and it is used by the server to verify and remove protection.

For the use of the various elements see Table 22 and Table 24.

4.4.9.2.6 required_protection

Stipulates the required protection on the attribute values / invocation and return parameters of methods accessed through the “Data protection” object.

enum:

When the enum value is interpreted as an unsigned, the meaning of each bit is as shown below:

Bit	Required protection
0	unused, shall be set to 0,
1	unused, shall be set to 0,
2	authenticated request,
3	encrypted request,
4	digitally signed request,
5	authenticated response,
6	encrypted response,
7	digitally signed response

4.4.9.3 Method description

4.4.9.3.1 get_protected_attributes (data)

Gets the value of the attributes specified by the object_list element, with the protection according to the protection_parameters in get_protected_ attributes_response applied.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Method invocation parameters:

```
data::= get_protected_attributes_request

get_protected_attributes_request::= structure
{
    object_list:            array object_definition,
    protection_parameters:  array protection_parameters_element,
}
```

Return parameters:

```
data::= get_protected_attributes_response

get_protected_attributes_response::= structure
{
    protection_parameters: array protection_parameters_element,
    protected_attributes:  octet-string
}
```

Where:

– object_list: see the *protection_object_list* attribute (4.4.9.2.3);

– protection_parameters::= array protection_parameters_element.

protection_parameters_element: see the *protection_parameters_get* attribute (4.4.9.2.4).

The protection_parameters in get_protected_attributes_response are elaborated by copying the protection_parameters from get_protected_attributes_request except that the server may need to fill in some elements. The remote party uses the protection_parameters in get_protected_attributes_response to verify / remove protection on the protected_attributes returned.

For the use of the various elements see Table 22 and Table 25.

4.4.9.3.2 set_protected_attributes (data)

Sets the value of the attributes specified by the object_list element, after verifying and removing the protection on the protected_attributes element according to the protection_parameters element.

Method invocation parameters:

```
data::= set_protected_attributes_request
set_protected_attributes_request::= structure
{
    object_list:            array object_definition,
    protection_parameters:  array protection_parameters_element,
    protected_attributes:   octet-string
}
```

Where:

– object_list: see the *protection_object_list* attribute (4.4.9.2.3);

– protection_parameters::= array protection_parameters_element.

protection_parameters_element: see the *protection_parameters_get* attribute (4.4.9.2.4).

For the use of the various elements see Table 22 and Table 26.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9.3.3 invoke_protected_method (data)

Invokes the method specified by the object_method element, after verifying and removing the protection on protected_method_invocation_parameters according to the protection_parameters in invoke_protected_method_ request.

After invoking the method specified by the object_method element, the protection according to the protection_parameters in invoke_protected_ method_response – that shall meet *required_protection* – is applied on the return parameters, and invoke_protected_method_response composed of protection_parameters and protected_method_return_parameters is returned.

Method invocation parameters:

```
data::= invoke_protected_method_request

invoke_protected_method_request::=     structure
{
      object_method:                          object_method_definition,

      protection_parameters: array            protection_parameters_element,
      protected_method_invocation_parameters:       octet-string
}

object_method_definition::=   structure
{
      class_id:          long-unsigned,
      logical_name:      octet-string,
      method_index:      integer,
}
```

Return parameters:

```
data::= invoke_protected_method_response

invoke_protected_method_response::= structure
{
      protection_parameters: array            protection_parameters_element,
      protected_method_return_parameters:     octet-string
}
```

protection_parameters_element: see the specification of the protection_parameters_get attribute (4.4.9.2.4).

If the method invoked does not provide return parameters then invoke_protected_method_response shall be a structure, with protection_parameters an array of zero elements and protected_method_return_parameters an octet-string of zero length.

To verify and remove protection from protected_method_invocation_parameters, the server uses the protection_parameters in invoke_protected_method_ request that shall meet *required_protection*.

The protection_parameters in the invoke_protected_method_response are elaborated by copying the protection_parameters from invoke_protected_method_request to invoke_protected_method_response except that the server may need to fill in some elements.

The protection_parameters in invoke_protected_method_response – that shall meet *required_protection* – are then applied to protect data in the protected_method_return_parameters.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9.4 Key_information and protection parameters

For the use of the various elements see Table 22 and Table 27.

Table 22 – Key information required to establish data protection keys

Key information choices		Comment
key_info_options		
(0) **identified_key_info_options**	S	The EK is identified
(0) global_unicast_encryption_key	S	GUEK
(1) global_broadcast_encryption_key	S	GBEK
(1) **wrapped_key_info_options**	S	The EK is transported using key wrap
kek_id	M	
(0) master_key	M	Identifies the key used for wrapping the key_ciphered_data. 0 = Master Key (KEK)
key_ciphered_data	M	Randomly generated key wrapped with KEK.
(2) **agreed_key_info_options**	S	The key is agreed by the parties using either: – the One-Pass Diffie-Hellman C(1e, 1s, ECC CDH) scheme or – the Static Unified Model C(0e, 2s, ECC CDH) scheme
key_parameters	M	Identifier of the key agreement scheme: 0x01: C(1e, 1s, ECC CDH) 0x02: C(0e, 2s, ECC CDH) All other reserved.
key_ciphered_data	M	– In the case of the C(1e, 1s, ECC CDH) scheme: the public key Qu of the ephemeral key agreement key pair of party U, signed with the private digital signature key of party U. – In the case of the C(0e, 2s, ECC CDH) scheme: an octet-string of length zero. NOTE In the second case party U has to provide a nonce, NonceU. See IEC 62056-5-3:2023, 5.3.4.6.4 and 5.3.4.6.5.
M: Mandatory (part of a structure) S: Selectable (part of a CHOICE)		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 23 – Protection parameters of *protection_parameters_get* attribute

protection_parameters_get		Written by the client	Filled in by the server
array protection_parameters_element	M	x	
protection_type	M	x	
(0) authentication	S	x	
(1) encryption	S	x	
(2) authentication and encryption	S	x	
(3) digital signature	S	x	
protection_options	M	x	
transaction_id	M	x	
originator_system_title (Shall carry the server system title)	M	x	
recipient_system_title (Shall carry the remote party system title)	M	x	
other_information	M	x	
key_info	M	x	
key_info_type	M	x	
(0) identified_key	S	x	
(1) wrapped_key	S	x	
(2) agreed_key	S	x	
key_info_options	M	x	
identified_key_options	S	x	
(0) global_unicast_encryption_key	S	x	
(1) global_broadcast_encryption_key	S	x	
wrapped_key_options	S	x	
kek_id	M	x	
(0) master_key	M	x	
key_ciphered_data	M	x	
agreed_key_options	S	x	
key_parameters	M	x	
key_ciphered_data	M		x
The *protection_parameters_get* attribute is written by the client, but when key_info_type is (2) agreed key, the key_ciphered_data of agreed_key_options is empty. When the One-pass Diffie-Hellman c(1e, 1s, ECC CDH) scheme is used, this element is filled by the server and the remote party has to read back *protection_parameters_get* in order to be able to verify and remove protection.			
M: Mandatory (part of a structure). S: Selectable (part of a CHOICE).			

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 24 – Protection parameters of *protection_parameters_set* attribute

protection_parameters_set		Written by the client	Filled in by the server
array protection_parameters_element	M	x	
protection_type	M	x	
(0) authentication	S	x	
(1) encryption	S	x	
(2) authentication and encryption	S	x	
(3) digital signature	S	x	
protection_options	M	x	
transaction_id	M	x	
originator_system_title (Shall carry the remote party system title)	M	x	
recipient_system_title ((Shall carry the server systems title)	M	x	
other_information	M	x	
key_info	M	x	
key_info_type	M	x	
(0) identified_key	S	x	
(1) wrapped_key	S	x	
(2) agreed_key	S	x	
key_info_options	M	x	
identified_key_options	S	x	
(0) global_unicast_encryption_key	S	x	
(1) global_broadcast_encryption_key	S	x	
wrapped_key_options	S	x	
kek_id	M	x	
(0) master_key	M	x	
key_ciphered_data	M	x	
agreed_key_options	S	x	
key_parameters	M	x	
key_ciphered_data	M	x	
M: Mandatory (part of a structure). S: Selectable (part of a CHOICE).			

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 25 – Protection parameters of *get_protected_attributes* method

Protection parameters	request	response
array protection_parameters_element	M	M (=)
protection_type	M	M (=)
(0) authentication	S	S (=)
(1) encryption	S	S (=)
(2) authentication and encryption	S	S (=)
(3) digital signature	S	S (=)
protection_options	M	M (=)
transaction_id	M	M (=)
originator_system_title	M	M[1]
recipient_system_title	M	M[2]
other_information	M	M (=)
key_info	M	M (=)
key_info_type	M	M (=)
(0) identified_key	S	S (=)
(1) wrapped_key	S	S (=)
(2) agreed_key	S	S (=)
key_info_options	M	M (=)
identified_key_options	S	S (=)
(0) global_unicast_encryption_key	S	S (=)
(1) global_broadcast_encryption_key	S	S (=)
wrapped_key_options	S	S (=)
kek_id	M	M (=)
(0) master_key	M	M (=)
key_ciphered_data	–	M[3]
agreed_key_options	S	S (=)
key_parameters	M	M (=)
key_ciphered_data	–	M[4]

Notice that protection parameters are taken from request and put into response. Protection parameters are only applied for protection of response.

[1] originator_system_title from request is put into recipient_system_title of response.

[2] recipient_system_title from request is put into originator_system_title of response.

[3] key_ciphered_data is filled by the server.

[4] When the One-pass Diffie-Hellman C(1e, 1s, ECC CDH) scheme is used, this element is filled by the server.

M: Mandatory (part of a structure).

S: Selectable (part of a CHOICE).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 26 – Protection parameters of *set_protected_attributes* method

Protection parameters	request	response
array protection_parameters_element	M	
protection_type	M	
(0) authentication	S	
(1) encryption	S	
(2) authentication and encryption	S	
(3) digital signature	S	
protection_options	M	
transaction_id	M	
originator_system_title	M	
recipient_system_title	M	
other_information	M	
key_info	M	
key_info_type	M	
(0) identified_key	S	
(1) wrapped_key	S	
(2) agreed_key	S	
key_info_options	M	
identified_key_options	S	
(0) global_unicast_encryption_key	S	
(1) global_broadcast_encryption_key	S	
wrapped_key_options	S	
kek_id	M	
(0) master_key	M	
key_ciphered_data	M	
agreed_key_options	S	
key_parameters	M	
key_ciphered_data	M	
Notice that protection parameters are taken from request and are not present in response. Protection parameters are only used for removing protection on the request.		
M: Mandatory (part of a structure). S: Selectable (part of a CHOICE).		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 27 – Protection parameters of *invoke_protected_method* method

Protection parameters	request	response
array protection_parameters_element	M	M(=)
protection_type	M	M(=)
(0) authentication	S	S(=)
(1) encryption	S	S(=)
(2) authentication and encryption	S	S(=)
(3) digital signature	S	S(=)
protection_options	M	M(=)
transaction_id	M	M(=)
originator_system_title	M	M[1]
recipient_system_title	M	M[2]
other_information	M	M(=)
key_info	M	M(=)
key_info_type	M	M(=)
(0) identified_key	S	S(=)
(1) wrapped_key	S	S(=)
(2) agreed_key	S	S(=)
key_info_options	M	M(=)
identified_key_options	S	S(=)
(0) global_unicast_encryption_key	S	S(=)
(1) global_broadcast_encryption_key	S	S(=)
wrapped_key_options	S	S(=)
kek_id	M	M(=)
(0) master_key	M	M(=)
key_ciphered_data	M	M[3]
agreed_key_options	S	S(=)
key_parameters	M	M(=)
key_ciphered_data	M	M[4]

Notice that protection parameters are taken from request and put into response. Protection parameters in the request are used to remove protection from request and protection parameters in the response are used to apply protection on response.

[1] originator_system_title from request is put into recipient_system_title of response.

[2] recipient_system_title from request is put into originator_system_title of response.

[3] key_ciphered_data is sent by the originator in the request and filled by the server for the response.

[4] When the One-pass Diffie-Hellman C(1e, 1s, ECC CDH) scheme is used, this element is filled by the server.

M: Mandatory (part of a structure).

S: Selectable (part of a CHOICE).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.10 Function control (class_id: 122, version: 0)

4.4.10.1 Overview

Instances of the IC “Function control” allow enabling and disabling functions in the server. Each function that can be enabled / disabled is identified by a name and is defined by a particular set of object identifiers referenced.

To allow enabling and disabling of functions controlled by time, “Single action schedule” and “Script table” objects are also specified.

Function control		0...n	class_id = 122, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. activation_status	(dyn.)	array				x + 0x08
3. function_list	(static)	array				x + 0x10
Specific methods		***m/o***				
1. set_function_status (data)		m				x + 0x20
2. add_function (data)		o				x + 0x28
3. remove_function (data)		o				x + 0x30

4.4.10.2 Attribute description

4.4.10.2.1 logical_name

Identifies the “Function control” object instance. See 6.2.38.

4.4.10.2.2 activation_status

Shows the current status of each functional block defined in the *function_list* attribute.

```
activation_status::= array function_status_type
function_status_type::= structure
{
        function_name    octet-string,
        function_status  boolean
}

        TRUE:    function enabled,

        FALSE:   function disabled.
```

4.4.10.3 function_list

Defines a list of functions which can be enabled or disabled by invoking the *set_function_status* method.

```
function_list::= array    functional_block

functional_block::=       structure
{
        function_name:          octet-string,
        function_specification: array function_definition
}

function_definition::= structure
{
        class_id:               long-unsigned,
        logical_name:           octet-string
}
```

The function_specification element contains a list of objects involved in this function. The objects are referenced by their class ids and logical names.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

In case of functions that cannot be linked to specific objects, class_id = 0 is used and the second element holds a descriptive name instead of a logical_name.

The names of the functions and the behaviour of the server in case of enabling or disabling a function have to be defined in project specific companion specifications.

Please also note that this attribute controls only the functions that are available in the server and that can be enabled or disabled. It is not possible to download new functions to the server by modifying this attribute or by invoking the *add_function* method.

4.4.10.4 Method description

4.4.10.4.1 set_function_status (data)

Enables or disables one or more functions defined in attribute *function_list*.

data::= array function_status_type

as defined in attribute *activation_status* (4.4.10.2.2).

The status of functions that are not listed is not modified.

4.4.10.4.2 add_function (data)

Adds a new function to the attribute *function_list*.

If a function with the same name already exists, the existing function will be replaced by the new function.

data::= functional_block

functional_block see 4.4.10.3.

4.4.10.4.3 remove_function (data)

Removes a function from the attribute *function_list*.

data::= octet-string, holding the function_name.

4.4.11 Array manager (class_id = 123, version = 0)

4.4.11.1 Overview

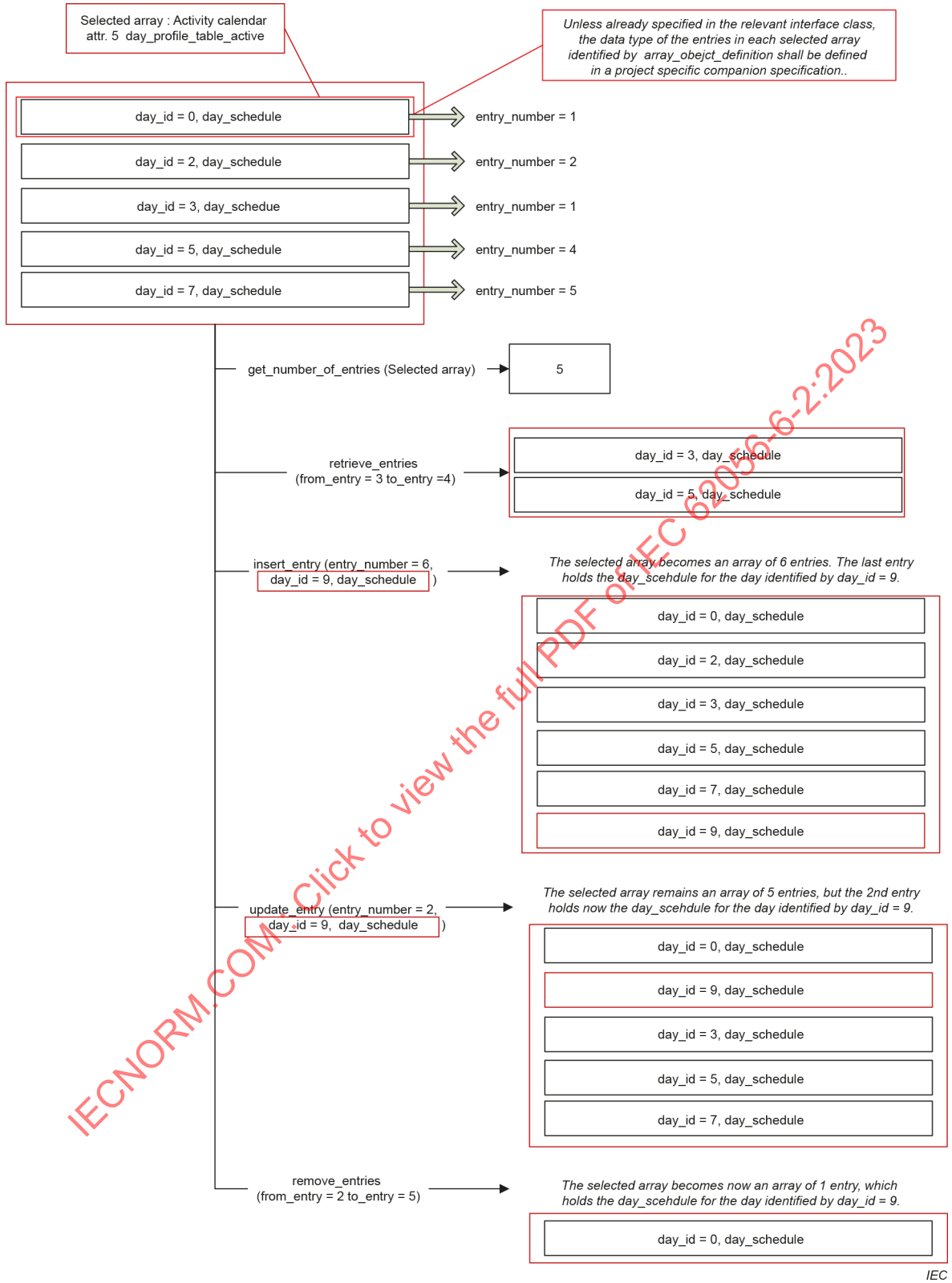
Instances of the “Array manager” IC allow managing attributes of type *array* of other interface objects, i.e.:

- retrieving the number of entries;
- selectively reading a range of entries;
- inserting a new entry or updating an existing entry;
- removing a range of entries.

Each instance allows managing several attributes of type *array* assigned to it.

An example of the application is shown in Figure 16.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 16 – Example of managing an array

Array manager		0...n	class_id = 123, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. array_object_list	(static)	array				x + 0x08
Specific methods		***m/o***				
1. retrieve_number_of_entries(data)		m				x + 0x20
2. retrieve_entries (data)		m				x + 0x28
3. insert_entry (data)		o				x + 0x30
4. update_entry (data)		o				x + 0x38
5. remove_entries (data)		o				x + 0x40

4.4.11.2 Attribute description

4.4.11.2.1 logical_name

Identifies the “Array manager” object instance. See 6.2.16.

4.4.11.2.2 array_object_list

Defines the list of attributes of type array that can be managed by an instance of this IC. Each “Array manager” object can manage 1 to n arrays.

```
array        array_object_list_definition

array_object_list_definition::=        structure
{
        array_object_id:                    unsigned,
        array_object_list_element:          array_object_definition
}
array_object_definition::=      structure
{
        class_id:           long-unsigned,
        logical_name:       octet-string,
        attribute_index:    integer
}
```

Where attribute_index is a pointer to the attribute within the object identified by its class_id and logical_name. Attribute_index 1 refers to the 1st attribute (i.e. the logical_name), attribute_index 2 to the 2nd, etc.

Each attribute identified by the array_object_definition shall be of type array.

An attribute of type array holds a number of entries that can be referenced by their entry number. The number 1 identifies the first entry and the number m identifies the last entry, where m is the number of entries of a selected array.

Unless already specified in the relevant interface class, the data type of the entries in each array identified by array_object_definition shall be specified in a project specific companion specification.

When the target array is accessed through an “Array manager” object, its access rights are observed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.11.3 Method description

4.4.11.3.1 retrieve_number_of_entries (data)

Returns the number of entries in the array identified.

The method invocation parameter identifies an array from the array_object_list by its array_object_id:

data::= unsigned

The return parameter holds the number of entries in the array identified.

4.4.11.3.2 retrieve_entries (data)

Returns a range of entries in one of the arrays of the array_object_list.

The method invocation parameters identify an array from the array_object_list by its array_object_id and the entries to be retrieved:

```
data::= entries_to_retrieve

entries_to_retrieve::=   structure
{
        array_object_id:  unsigned,
        list_of_entries:   structure
        {
                from_entry:        long-unsigned,
                to_entry:          long-unsigned
        }
}
```

Where:

– array_object_id identifies one of the arrays managed by an “Array manager” object;

– list_of_entries identifies the entries to be retrieved.

If the number of entries in the selected array is m:

– if from_entry < last entry < m, then the entries in this range are retrieved;

– if from_entry < last entry but last_entry > m, then the entries identified by from_entry up to the last entry are retrieved;

– if from_entry < last_entry, but from_entry > m, then the method returns an array of 0 elements;

– if from_entry > last_entry the method invocation fails.

The return parameters hold an array of the entries selected:

```
data::=       retrieved_entries:

retrieved_entries:        array entry

entry::= attribute specific
```

The data type depends on the attribute of type *array* as specified in the IC specification or in the project specific companion specification.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.11.3.3 insert_entry (data)

Inserts a new entry in a selected array.

```
data::=          structure
{
    array_object_id:      unsigned,
    entry_to_insert:      structure
        {
            entry_number:   long-unsigned,
            entry:          attribute specific
```

The data type of the entry depends on the attribute of type *array* as specified in the IC specification or in the project specific companion specification.

```
        }
}
```

Where:

– array_object_id identifies one of the arrays managed by an “Array manager” object;

– entry_to_insert identifies the entry_number and holds the entry itself to be inserted.

If entry_number = 0:

– if the selected array is already full, then the method invocation fails;

– else, the new entry is inserted at the beginning of the array: it becomes the first entry and all the other entries of the selected array are shifted up by one;

If entry_number > m (where m is the number of entries in the array):

– if the selected array is already full, then the method invocation fails;

– else, the new entry is inserted at the end of the selected array: it becomes the last entry. The other entries are not shifted.

If an entry_number identifies an existing entry, it is inserted after the existing entry.

If the entries are shifted due to inserting a new entry then all the references to them have to be updated.

4.4.11.3.4 update_entry (data)

Updates an existing entry in a selected array.

```
data::=          structure
{
    array_object_id:      unsigned,
    entry_to_update:      structure
        {
        entry_number:   long-unsigned,
        entry:          attribute specific
```

The data type of the entry depends on the attribute of type *array* as specified in the IC specification or in the project specific companion specification.

```
        }
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

– array_object_id identifies one of the arrays managed by an “Array manager” object;
– entry_to_update identifies the entry_number and holds the entry itself to be updated.

When an entry_number identifies an existing entry, it is overwritten.

In any other case, the method invocation fails.

4.4.11.3.5 remove_entries (data)

Removes a range of entries in one of the arrays of the array_object_list.

The method invocation parameters identify an array from the array_object_list by its array_object_id and the entries to be removed:

```
data::= entries_to_remove

entries_to_remove::=   structure
{
      array_object_id:  unsigned;
      list_of_entries:  structure
      {
            from_entry:       long-unsigned,
            to_entry:         long-unsigned
      }
```

Where:

– array_object_id identifies one of the arrays managed by an “Array manager” object;
– list_of_entries identifies the entries to be removed.

If the number of entries in the selected array is m:

– if from_entry < last entry< m, then the entries in this range are removed;
– if from_entry < last entry but last_entry > m, then the entries identified by from_entry up to the last entry are removed;
– if from_entry < last_entry, but from_entry > m, then no entries are removed;
– if from_entry > last_entry the method invocation fails.

If the entries are shifted due to removing entries, then all the references to them have to be updated.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.12 Communication port protection (class_id = 124, version = 0)

4.4.12.1 Overview

Instances of the “Communication port protection” IC can be used to protect communication ports of DLMS®/COSEM servers against possibly malicious communication attempts, in particular to prevent brute force attacks by reducing the possible number of attempts.

Each instance references a single communication port by its logical name (OBIS code). If an acceptable number of failed attempts is exceeded then the communication port is temporarily locked. The lockout time may increase with each failed attempt, until a maximum lockout time is reached.

A failed attempt is one that leads to discarding the APDU carrying a service request. The criteria for detecting a failed attempt are out of the Scope of this document.

The objects count both the number of failed attempts between two resets and the cumulative number of failed attempts.

It is possible to configure the communication ports such that they are locked to all attempts or unlocked to all attempts

Communication port protection		**0...n**	**class_id = 124, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name	(static)	octet-string				x
2. protection_mode	(static)	enum			1	x + 0x08
3. allowed_failed_attempts	(static)	long-unsigned				x + 0x10
4. initial_lockout_time	(static)	double-long-unsigned				x + 0x18
5. steepness_factor	(static)	unsigned			1	x + 0x20
6. max_lockout_time	(static)	double-long-unsigned				x + 0x28
7. port_reference	(static)	octet-string				x + 0x30
8. protection_status	(dyn.)	enum				x + 0x38
9. failed_attempts	(dyn.)	double-long-unsigned				x + 0x40
10. cumulative_failed_attempts	(dyn.)	double-long-unsigned				x + 0x48
Specific methods	***m/o***					
1. reset (data)	o	o				x + 0x50

4.4.12.2 Attribute description

4.4.12.2.1 logical_name

Identifies the “Communication port protection” object instance. See 6.2.39.

4.4.12.2.2 protection_mode

Controls the protection mode.

enum

(0) locked,
(1) locked_on_failed_attempts,
(2) unlocked

When set to locked, no communication can take place via the port referenced by attribute 7.

When set to “locked_on_failed_attempts”, the port becomes temporarily locked when the number of failed communication attempts exceeds an allowed number. The length of the lockout period may increase with each failed attempt, as determined by attributes 5 and 6.

When set to “unlocked” the lockout mechanism is disabled; communication through the port referenced by attribute 7 is always possible.

When the port is locked no communication can take place, even valid APDUs are discarded.

4.4.12.2.3 allowed_failed_attempts

Holds the number of allowed failed communication attempts before the lockout mechanism is triggered.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.12.2.4 initial_lockout_time

Holds the initial value of the lockout time, in seconds, after the first failed communication attempt when the value of allowed_failed_attempts is reached.

4.4.12.2.5 steepness_factor

Holds a factor that controls how the lockout time is increased with each failed attempt when allowed_failed_attempts is reached, until the max_lockout_time is reached.

The current lockout time is calculated using the following formulae:

$$NCA = failed_attempts - allowed_failed_attempts$$

$$CLT = initial_lockout_time \times (steepness_factor^{(NCA-1)})$$

Where:

– NCA is number of counted failed attempts: that is: the number of attempts that count towards extending the lockout time;

– CLT is the current lockout time.

Table 28 shows example values with an initial lockout time of 60 s.

Table 28 – Example values for NCA and CLT

NCA	steepness_factor		
	1	2	3
	CLT (in seconds)		
1	60	60	60
2	60	120	180
3	60	240	540
4	60	480	1 620
5	60	960	4 860

Once lockout occurs, even valid attempts are discarded until the current lockout time has expired.

4.4.12.2.6 max_lockout_time

Holds the maximum time, in seconds, for which the communication port can be locked, even if the number of failed attempts keeps increasing. The purpose of this attribute is to avoid a denial of service attack.

4.4.12.2.7 port_reference

Contains the logical name of an object that identifies the communication port being protected. That object may be a communication port setup object or any other suitably chosen object.

If this information is not available or not needed, the attribute may be left empty (octet-string [0]).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.12.2.8 protection_status

Holds the current protection status of the communication port.

enum

(0) unlocked,
(1) temporarily_locked,
(2) locked

The status is unlocked if the protection_mode is unlocked or the number of failed attempts does not exceed the number of allowed_failed_attempts.

NOTE 1 The status becomes unlocked again when the current_lockout_time has elapsed.

The status is temporarily_locked if the number of failed communication attempts exceeds allowed_failed_attempts and the current_lockout_time has not yet expired.

The status is locked if the protection_mode is locked.

NOTE 2 When a port is locked, the server still may be accessed on another port.

failed_attempts Holds the total number of failed attempts since the last reset, independent of the protection_mode and the triggering of the protection mechanism.

4.4.12.2.9 cumulative_failed_attempts

Holds the cumulative number of failed attempts, independent of the protection_mode and the triggering of the protection mechanism.

This attribute is never reset.

4.4.12.3 Method description

4.4.12.3.1 reset (data)

Resets the variables of the lockout mechanism:

– the value of failed_attempts is reset to 0;
– the current lockout time is reset to 0;
– the protection_status is set to “(0) unlocked” if the protection_mode is (1) locked_on_failed_attempts, and is not affected otherwise.

data::= integer (0)

4.5 Interface classes for time- and event bound control

4.5.1 Clock (class_id = 8, version = 0)

4.5.1.1 Overview

This IC models the device clock, managing all information related to date and time including deviations of the local time to a generalized time reference (UTC) due to time zones and daylight saving time schemes. The IC also offers various methods to adjust the clock.

The *date* information includes the elements year, month, day of month and day of week. The *time* information includes the elements hour, minutes, seconds, hundredths of seconds, and the deviation of the local time from UTC. The daylight saving time function modifies the deviation of local time to UTC depending on the attributes; see Figure 17.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The start and end point of that function is normally set once. An internal algorithm calculates the real switch point depending on these settings.

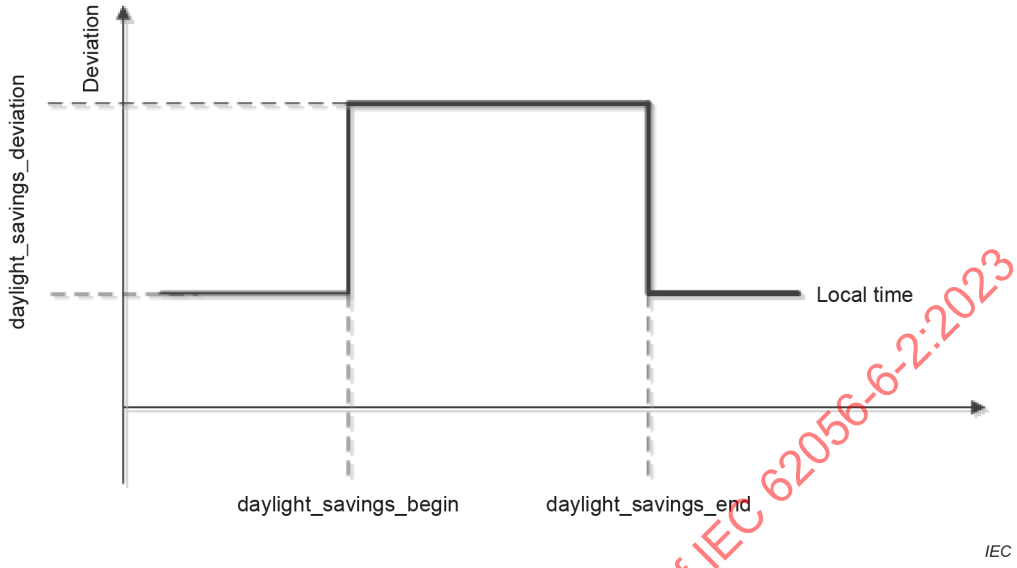


Figure 17 – The generalized time concept

Clock		**0...n**	**class_id = 8, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. time	(dyn.)	octet-string				x + 0x08
3. time_zone	(static)	long	-720	+840		x + 0x10
4. status	(dyn.)	unsigned				x + 0x18
5. daylight_savings_begin	(static)	octet-string				x + 0x20
6. daylight_savings_end	(static)	octet-string				x + 0x28
7. daylight_savings_deviation	(static)	integer	-120	+120		x + 0x30
8. daylight_savings_enabled	(static)	boolean				x + 0x38
9. clock_base	(static)	enum				x + 0x40
Specific methods		***m/o***				
1. adjust_to_quarter (data)		o				x + 0x60
2. adjust_to_measuring_period (data)		o				x + 0x68
3. adjust_to_minute (data)		o				x + 0x70
4. adjust_to_preset_time (data)		o				x + 0x78
5. preset_adjusting_time (data)		o				x + 0x80
6. shift_time (data)		o				x + 0x88

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.1.2 Attribute description

4.5.1.2.1 logical_name

Identifies the “Clock” object instance. See 6.2.5.

4.5.1.2.2 time

Contains the meter's local date and time, its deviation to UTC and the status.

octet-string, formatted as specified in 4.1.6.1 for *date-time.*

When this attribute is set, all the fields in the octet-string representing *date-time* shall be evaluated and the local date and time in the meter shall be set according to the rules defined in 4.1.6.1.

Only specified fields of the *date-time* are changed.

EXAMPLE For setting the *date* without changing the *time*, all time-relevant octets in the octet string representing *date-time* shall be set to "not specified".

4.5.1.2.3 time_zone

The deviation of local, normal time to UTC in minutes. The value depends on the geographical location of the meter.

4.5.1.2.4 status

The clock_status maintained by the meter. The clock_status element indicated in the *time* attribute is equal to the value of this attribute.

unsigned, formatted as specified in 4.1.6.1 for *clock_status*

4.5.1.2.5 daylight_savings_ begin

Defines the local switch date and time when the local time starts to deviate from the normal time.

For generic definitions, wildcards are allowed.

octet-string, formatted as specified in 4.1.6.1 for *date-time.*

4.5.1.2.6 daylight_savings_end

Defines the local switch date and time when the local time ends to deviate from the local normal time.

octet-string, formatted as specified in 4.1.6.1 for *date-time.*

4.5.1.2.7 daylight_savings_ deviation

Contains the number of minutes by which the deviation in generalized time shall be corrected at daylight savings begin.

integer: Deviation in the range of ± 120 min

4.5.1.2.8 daylight_savings_ enabled

boolean: TRUE: DST enabled,
FALSE: DST disabled

NOTE This is a parameter to enable and disable the daylight savings time feature. The current status is indicated in the status attribute.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.1.2.9 clock_base

Defines where the basic timing information comes from.

enum:

(0) not defined,
(1) internal crystal,
(2) mains frequency 50 Hz,
(3) mains frequency 60 Hz,
(4) GPS (global positioning system),
(5) radio controlled.

4.5.1.3 Method description

4.5.1.3.1 adjust_to_quarter (data)

Sets the meter's time to the nearest (+/-) quarter of an hour value (*:00, *:15, *:30, *:45).

data::= integer (0)

4.5.1.3.2 adjust_to_measuring_period (data)

Sets the meter's time to the nearest (+/-) starting point of a measuring period.

data::= integer (0)

4.5.1.3.3 adjust_to_minute (data)

Sets the meter's time to the nearest minute.

If second_counter < 30 s, second_counter is set to 0.

If second_counter ≥ 30 s, second_counter is set to 0, and minute_counter and all depending clock values are incremented if necessary.

data::= integer (0)

4.5.1.3.4 adjust_to_preset_ time (data)

This method is used in conjunction with the preset_adjusting_time method. If the meter's time lies between validity_interval_start and validity_interval_end, then time is set to preset_time.

data::= integer (0)

4.5.1.3.5 preset_adjusting_ time (data)

Presets the time to a new value (preset_time) and defines a validity_interval within which the new time can be activated.

```
data::= structure
{
        preset_time:                octet-string,
        validity_interval_start:    octet-string,
        validity_interval_end:      octet-string
}
```

all octet-strings formatted as specified in 4.6.1 for *date-time*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.1.3.6 shift_time (data)

Shifts the time by *n* (-900 <= *n* <= 900) s.

data::= long

4.5.2 Script table (class_id = 9, version = 0)

4.5.2.1 Overview

This IC allows modelling the triggering of a series of actions by executing scripts using the *execute* (data) method.

"Script table" objects contain a table of script entries. Each entry consists of a script identifier and a series of action specifications. An action specification activates a method or modifies an attribute of a COSEM object within the logical device.

A certain script may be activated by other COSEM objects within the same logical device or from the outside.

If two scripts have to be executed at the same time instance, then the one with the smaller index is executed first.

Script table	**0...n**	**class_id = 9, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. scripts (static)	array				x + 0x08
Specific methods	***m/o***				
1. execute (data)	m				x + 0x20

4.5.2.2 Attribute description

4.5.2.2.1 logical_name

Identifies the "Script table" object instance. See 6.2.7.

4.5.2.2.2 scripts

Specifies the different scripts, i.e. the lists of actions.

```
array        script

script::= structure
{
        script_identifier:  long-unsigned,
        actions:            array action_specification
}
```

The script_identifier 0 is reserved. If specified with an *execute* method, it results in a null script (no actions to perform).

```
action_specification::= structure
{
        service_id:             enum,
        class_id:               long-unsigned,
        logical_name:           octet-string,
        index:                  integer,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

parameter: service specific
}

Where:

– the service_id element defines which action to be applied to the referenced object:
  (1) write attribute,
  (2) execute specific method
– the index element defines (with service_id 1) which attribute of the selected object is affected; or (with service_id 2) which specific method is to be executed. The first attribute (logical_name) has index 1, the first specific method has index 1 as well.

NOTE 1 The action_specification is limited to activate methods that do not produce any response (from the server to the client).

NOTE 2 A “dummy” action specification with all elements 0 means that the action is not configured.

4.5.2.3 Method description

4.5.2.3.1 execute (data)

Executes the script specified in parameter data.

data::= long-unsigned

If data matches one of the script_identifiers in the script table, then the corresponding action_specification is executed.

4.5.3 Schedule (class_id = 10, version = 0)

4.5.3.1 Overview

This IC, together with the IC “Special days”, allows modelling time- and date-driven activities within a device. Table 29 and Table 30 provide an overview and show the interactions between the two ICs.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 29 – Schedule

Index	enable	action (script)	Switch _time	validity _ window	exec_weekdays							exec_specdays					date range	
					Mo	Tu	We	Th	Fr	Sa	Su	S1	S2	...	S8	S9	begin_ date	end_ date
120	Yes	xxxx:yy	06:00	0xFFFF	x	x	x	x	x	x							xx-04-01	xx-09-30
121	Yes	xxxx:yy	22:00	15	x	x	x	x	x								xx-04-01	xx-09-30
122	Yes	xxxx:yy	12:00	0						x							xx-04-01	xx-09-30
200	No	xxxx:yy	06:30		x	x	x	x	x	x							xx-04-01	xx-09-30
201	No	xxxx:yy	21:30		x	x	x	x	x								xx-04-01	xx-09-30
202	No	xxxx:yy	11:00							x							xx-04-01	xx-09-30

Table 30 – Special days table

Index	special_day_date	day_id
12	xx-12-24	S1
33	xx-12-25	S3
77	97-03-31	S3

Schedule		0...n	class_id = 10, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. entries	(static)	array				x + 0x08
Specific methods		***m/o***				
1. enable/disable (data)		o				x + 0x20
2. insert (data)		o				x + 0x28
3. delete (data)		o				x + 0x30

4.5.3.2 Attribute description

4.5.3.2.1 logical_name

Identifies the “Schedule” object instance. See 6.2.9.

4.5.3.2.2 entries

Specifies the scripts to be executed at given times. There is only one script that can be executed per entry.

```
array        schedule_table_entry
schedule_table_entry::= structure
{
        index:                  long-unsigned,
        enable:                 boolean,
        script_logical_name:    octet-string,
        script_selector:        long-unsigned,
        switch_time:            octet-string,
        validity_window:        long-unsigned,
        exec_weekdays:          bit-string,
        exec_specdays:          bit-string,
        begin_date:             octet-string,
        end_date:               octet-string
}
```

Where:

– script_logical_name: defines the logical name of the “Script table” object;

– script_selector: defines the script_identifier of the script to be executed;

– switch_time accepts wildcards to define repetitive entries. The format of the octet-string follows the rules set in 4.1.6.1 for time;

– validity_window defines a period in minutes, in which an entry shall be processed after power fail. (time between defined switch_time and actual power_up) 0xFFFF: the script is processed any time;

– exec_weekdays defines the days of the week on which the entry is valid;

– exec_specdays perform the link to the IC “Special days table”, day_id;

begin_date and end_date define the date period in which the entry is valid (wildcards are allowed). The format follows the rules set in 4.1.6.1 for *date*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.3.3 Method description

4.5.3.3.1 enable/disable (data)

Sets the disabled bit of range A entries to true and then enables the entries of range B.

```
data::= structure
{
    firstIndexA:    long-unsigned,
    lastIndexA:     long-unsigned,
    firstIndexB:    long-unsigned,
    lastIndexB:     long-unsigned
}
```

Where:

– firstIndexA first index of the range that is disabled,

– lastIndexA last index of the range that is disabled,

– firstIndexB first index of the range that is enabled,

– lastIndexB last index of the range that is enabled,

– firstIndexA/B < lastIndexA/B: all entries of the range A/B are disabled / enabled,

– firstIndexA/B = lastIndexA/B: one entry is disabled/enabled,

– firstIndexA/B > lastIndexA/B: nothing disabled/enabled,

– firstIndexA/B and lastIndexA/B > 9999: no entry is disabled/enabled

4.5.3.3.2 insert (data)

Inserts a new entry in the table. If the index of the entry exists already, the existing entry is overwritten by the new entry.

```
entry: schedule_table_entry

data::=    corresponding to entry
```

4.5.3.3.3 delete (data)

Deletes a range of entries in the table.

```
data::= structure
{
    firstIndex:   long-unsigned,
    lastIndex:    long-unsigned
}
```

Where:

– firstIndex: first index of the range that is deleted,

– lastIndex: last index of the range that is deleted,

– firstIndex < lastIndex: all entries of the range A/B are deleted,

– firstIndex = lastIndex: one entry is deleted,

– firstIndex > lastIndex: nothing deleted

Remarks concerning “inconsistencies” in the table entries:

If the same script should be executed several times at a specific time instance, then it is executed only once.

If different scripts should be executed at the same time instance, then the execution order is according to the “index”. The script with the lowest “index” is executed first.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.3.4 Recovery after power failure

After a power failure, the whole schedule is processed to execute all the necessary scripts that would get lost during a power failure. For this, the entries that were not executed during the power failure have to be detected. Depending on the validity window they are executed in the correct order (as they would have been executed in normal operation).

4.5.3.5 Handling of time changes

There are four different "actions" of time changes:

a) time setting forward (by writing to the attribute *time* of the “Clock” object);

b) time setting backward (by writing to the attribute *time* of the “Clock” object);

c) time synchronization (using the method *adjust_to_quarter* of the “Clock” object);

d) daylight saving action.

All these four actions need a different handling executed by the schedule in interaction with the time setting activity.

4.5.3.6 Time setting forward

This is handled the same way as a power failure. All entries missed are executed depending on the validity_window. A (manufacturer specific defined) short time setting can be handled like time synchronization.

4.5.3.7 Time setting backward

This results in a repetition of those entries that are activated during the repeated time. A (manufacturer specific defined) short time setting can be handled like time synchronization.

4.5.3.8 Time synchronization

Time synchronization is used to correct small deviations between a master clock and the local clock. The algorithm is manufacturer specific. It shall guarantee that no entry of the schedule gets lost, or is executed twice. The validity_window attribute has no effect, because all entries have to be executed in normal operation.

4.5.3.9 Daylight saving

If the clock is put forward, then all scripts, which fall into the forwarding interval (and would therefore get lost) are executed. If the clock is put back, re-execution of the scripts, which fall into the backwarding interval, is suppressed.

4.5.4 Special days table (class_id = 11, version = 0)

4.5.4.1 Overview

This IC allows defining special dates. On such dates, a special switching behaviour overrides the normal one. The IC works in conjunction with the class "Schedule" or "Activity calendar". The linking data item is *day_id*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Special days table		0...n	class_id = 11, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. entries	(static)	array				x + 0x08
Specific methods		***m/o***				
1. insert (data)		o				x + 0x10
2. delete (data)		o				x + 0x18

4.5.4.2 Attribute description

4.5.4.2.1 logical_name

Identifies the “Special days table” object instance. See 6.2.8.

4.5.4.2.2 entries

Specifies a special day identifier for a given date. The date may have wildcards for repeating special days like Christmas.

```
array        spec_day_entry
spec_day_entry::= structure
{
        index:                  long-unsigned,
        specialday_date:        octet-string,
        day_id:                 unsigned
}
```

Where:

– specialday_date formatting follows the rules set in 4.6.1 for *date*;

– the range of the day_id shall match the length of the bit-string exec_specdays in the related object of IC “Schedule”.

4.5.4.3 Method description

4.5.4.3.1 insert (data)

Inserts a new entry in the table.

```
entry::= spec_day_entry
```

If a special day with the same index or with the same date as an already defined day is inserted, the old entry will be overwritten.

4.5.4.3.2 delete (data)

Deletes an entry in the table.

```
index       Index of the entry that shall be deleted.

data::= long-unsigned
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.5 Activity calendar (class_id = 20, version = 0)

4.5.5.1 Overview

This IC allows modelling the handling of various tariff structures in the meter. The IC provides a list of scheduled actions, following the classical way of calendar based schedules by defining seasons, weeks, etc.

An “Activity calendar” object may coexist with the more general “Schedule” object and it can even overlap with it. If actions in a “Schedule” object are scheduled for the same activation time as in an “Activity calendar” object, the actions triggered by the “Schedule” object are executed first.

After a power failure, only the “last action” missed from the object “Activity calendar” is executed (delayed). This is to ensure proper tariffication after power up. If a “Schedule” object is present, then the missed “last action” of the “Activity calendar” shall be executed at the correct time within the sequence of actions requested by the “Schedule” object.

The “Activity calendar” object defines the activation of certain scripts, which can perform different activities inside the logical device. The interface to the IC “Script table” is the same as for the IC “Schedule” (see 4.5.3).

If an instance of the IC “Special days table” (see 4.5.4) is available, relevant entries there take precedence over the “Activity calendar” object driven selection of a day profile. The day profile referenced in the “Special days table” activates the day_schedule of the *day_profile_table* in the “Activity calendar” object by referencing through the day_id.

Activity calendar		**0…n**	**class_id = 20, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. calendar_name_active	(static)	octet-string				x + 0x08
3. season_profile_active	(static)	array				x + 0x10
4. week_profile_table_active	(static)	array				x + 0x18
5. day_profile_table_active	(static)	array				x + 0x20
6. calendar_name_passive	(static)	octet-string				x + 0x28
7. season_profile_passive	(static)	array				x + 0x30
8. week_profile_table_passive	(static)	array				x + 0x38
9. day_profile_table_passive	(static)	array				x + 0x40
10. activate_passive_calendar_time	(static)	octet-string				x + 0x48
Specific methods		***m/o***				
1. activate_passive_calendar (data)		o				x + 0x50

4.5.5.2 Attribute description

NOTE Attributes called …_active are currently active. Attributes called …_passive will be activated by the specific method activate_passive_calendar.

4.5.5.2.1 logical_name

Identifies the “Activity calendar” object instance. See 6.2.10.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.5.2.2 calendar_name_active

Typically contains an identifier, which is descriptive to the set of scripts activated by the object.

4.5.5.2.3 season_profile_active

Contains a list of seasons defined by their starting date and a specific week_profile to be executed. The list is sorted according to season_start (in increasing order);

```
array        season

season::= structure
{
        season_profile_name:        octet-string,
        season_start:               octet-string,
        week_name:                  octet-string
}
```

Where:

– season_profile_name is a user defined name identifying the current season_profile;

– season_start defines the starting time of the season, formatted as specified in 4.6.1 for date-time. In season_start, wildcards are allowed. If all fields are wildcards, the season will never start. When using wildcards, special care has to be taken to avoid conflicting parametrization, i.e. that the season_start of two different seasons is the same;

NOTE The current season is terminated by the season_start of the next season.

– week_name defines the week_profile active in this season.

4.5.5.2.4 week_profile_table_active

Contains an array of week_profiles to be used in the different seasons. For each week_profile, the day_profile for every day of a week is identified.

```
array        week_profile

week_profile::= structure
{
        week_profile_name:    octet-string,
        monday:               day_id,
        tuesday:              day_id,
        wednesday:            day_id,
        thursday:             day_id,
        friday:               day_id,
        saturday:             day_id,
        sunday:               day_id
}

day_id: unsigned
```

Where:

– week_profile_name is a user defined name identifying the current week_profile;

– Monday defines the day_profile valid on Monday;

– …

Sunday defines the day_profile valid on Sunday.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.5.2.5 day_profile_table_active

Contains an array of day_profiles, identified by their day_id. For each day_profile, a list of scheduled actions is defined by a script to be executed and the corresponding activation time (start_time). The list is sorted according to start_time.

```
array        day_profile

day_profile::= structure
{
      day_id:            unsigned,
      day_schedule:      array day_profile_action
}

day_profile_action::= structure
{
      start_time:              octet-string,
      script_logical_name:     octet-string,
      script_selector:         long-unsigned
}
```

Where:

– day_id is a user defined identifier, identifying the current day_profile;

– start_time: defines the time when the script is to be executed (no wildcards); the format follows the rules set in 4.1.6.1 for *time*;

– script_logical_name: defines the *logical_name* of the "Script table" object;

– script_selector: defines the script_identifier of the script to be executed.

4.5.5.2.6 activate_passive_calendar_time

Defines the time when the object itself calls the specific method *activate_passive_calendar*. A definition with "not specified" notation in all fields of the attribute will deactivate this automatism. Partial "not specified" notation in just some fields of date and time are not allowed.

octet-string, formatted as specified in 4.6.1 for *date-time*.

4.5.5.2.7 calendar_name_passive

Typically contains an identifier, which is descriptive to the set of scripts activated by the object.

4.5.5.2.8 season_profile_passive

Contains a list of seasons defined by their starting date and a specific week_profile to be executed. The list is sorted according to season_start (in increasing order);

```
array        season

season::= structure
{
      season_profile_name:        octet-string,
      season_start:               octet-string,
      week_name:                  octet-string
}
```

Where:

– season_profile_name is a user defined name identifying the current season_profile;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– season_start defines the starting time of the season, formatted as specified in 4.1.6.1 for date-time. In season_start, wildcards are allowed. If all fields are wildcards, the season will never start. When using wildcards, special care has to be taken to avoid conflicting parametrization, i.e. that the season_start of two different seasons is the same;

NOTE The current season is terminated by the season_start of the next season.

– week_name defines the week_profile active in this season.

4.5.5.2.9 week_profile_table_passive

Contains an array of week_profiles to be used in the different seasons. For each week_profile, the day_profile for every day of a week is identified.

```
array        week_profile

week_profile::= structure
{
      week_profile_name:    octet-string,
      monday:               day_id,
      tuesday:              day_id,
      wednesday:            day_id,
      thursday:             day_id,
      friday:               day_id,
      saturday:             day_id,
      sunday:               day_id
}

day_id: unsigned
```

Where:

– week_profile_name is a user defined name identifying the current week_profile;

– Monday defines the day_profile valid on Monday;

– …

Sunday defines the day_profile valid on Sunday.

4.5.5.2.10 day_profile_table_passive

Contains an array of day_profiles, identified by their day_id. For each day_profile, a list of scheduled actions is defined by a script to be executed and the corresponding activation time (start_time). The list is sorted according to start_time.

```
array        day_profile

day_profile::= structure
{
      day_id:           unsigned,
      day_schedule:     array day_profile_action
}

day_profile_action::= structure
{
      start_time:             octet-string,
      script_logical_name:    octet-string,
      script_selector:        long-unsigned
}
```

Where:

– day_id is a user defined identifier, identifying the current day_profile;

– start_time: defines the time when the script is to be executed (no wildcards); the format follows the rules set in 4.1.6.1 for *time*;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– script_logical_name: defines the *logical_name* of the “Script table” object;
– script_selector: defines the script_identifier of the script to be executed.

4.5.5.2.11 activate_passive_calendar_time

Defines the time when the object itself calls the specific method *activate_passive_calendar*. A definition with "not specified" notation in all fields of the attribute will deactivate this automatism. Partial "not specified" notation in just some fields of date and time are not allowed.

octet-string, formatted as specified in 4.1.6.1 for *date-time*.

4.5.5.3 Method description

4.5.5.3.1 activate_passive_calendar (data)

This method copies all attributes called ..._passive to the corresponding attributes called ..._active.

data::= integer (0)

4.5.6 Register monitor (class_id = 21, version = 0)

4.5.6.1 Overview

This IC allows modelling the function of monitoring of values modelled by “Data”, “Register”, “Extended register” or “Demand register” objects. It allows specifying thresholds, the value monitored, and a set of scripts (see 4.5.2) that are executed when the value monitored crosses a threshold.

The IC “Register monitor” requires an instantiation of the IC “Script table” in the same logical device.

Register monitor		**0...n**	**class_id = 21, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. thresholds	(static)	array				x + 0x08
3. monitored_value	(static)	value_definition				x + 0x10
4. actions	(static)	array			0	x + 0x18
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.6.2 Attribute description

4.5.6.2.1 logical_name

Identifies the “Register monitor” object instance. See 6.2.13 and 6.3.10.

4.5.6.2.2 thresholds

Provides the threshold values to which the attribute of the referenced register is compared.

array threshold

threshold: The threshold is of the same type as the monitored attribute of the referenced object.

4.5.6.2.3 monitored_value

Defines which attribute of an object is to be monitored. Only values with simple data types are allowed.

```
value_definition::= structure
{
    class_id:           long-unsigned,
    logical_name:       octet-string,
    attribute_index:    integer
}
```

4.5.6.2.4 actions

Defines the scripts to be executed when the monitored attribute of the referenced object crosses the corresponding threshold. The attribute actions has exactly the same number of elements as the attribute thresholds. The ordering of the action_items correspond to the ordering of the *thresholds* (see 4.5.6.2.2).

```
array       action_set
action_set::=   structure
{
    action_up:      action_item,
    action_down:    action_item
}
```

Where:

– action_up defines the action when the attribute value of the monitored register crosses the threshold in the upwards direction;

– action_down defines the action when the attribute value of the monitored register crosses the threshold in the downwards direction.

```
action_item::= structure
{
    script_logical_name:    octet-string,
    script_selector:        long-unsigned
}
```

4.5.7 Single action schedule (class_id = 22, version = 0)

4.5.7.1 Overview

This IC allows modelling the execution of periodic actions within a meter. Such actions are not necessarily linked to tariffication (see “Activity calendar” (4.5.5) or “Schedule” (4.5.3)).

Single action schedule		**0...n**	**class_id = 22, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. executed_script	(static)	script				x + 0x08
3. type	(static)	enum				x + 0x10
4. execution_time	(static)	array				x + 0x18
Specific methods		***m/o***				

4.5.7.2 Attribute description

4.5.7.2.1 logical_name

Identifies the “Single action schedule” object instance. See 6.2.12.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.7.2.2 executed_script

Contains the logical name of the “Script table” object and the script selector of the script to be executed.

```
script::= structure
{
script_logical_name:    octet-string,
script_selector:        long-unsigned
}
```

Script_logical_name and script_selector define the script to be executed.

4.5.7.2.3 type

enum:

(1) size of execution_time = 1; wildcard in date allowed,
(2) size of execution_time = n; all time values are the same, wildcards in date not allowed,
(3) size of execution_time = n; all time values are the same, wildcards in date are allowed,
(4) size of execution_time = n; time values may be different, wildcards in date not allowed,
(5) size of execution_time = n; time values may be different, wildcards in date are allowed

4.5.7.2.4 execution_time

Specifies the time and the date when the script is executed.

```
array       execution_time_date

execution_time_date::= structure
{
        time: octet-string,
        date: octet-string
}
```

The two octet-strings contain *time* and *date*, in this order; *time* and *date* are formatted as specified in 4.1.6.1. Hundredths of seconds shall be zero.

4.5.8 Disconnect control (class_id = 70, version = 0)

4.5.8.1 Overview

Instances of the “Disconnect control” IC manage an internal or external disconnect unit of the meter (e.g. electricity breaker, gas valve) in order to connect or disconnect – partly or entirely – the premises of the consumer to / from the supply. The state diagram and the possible state transitions are shown in Figure 18.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

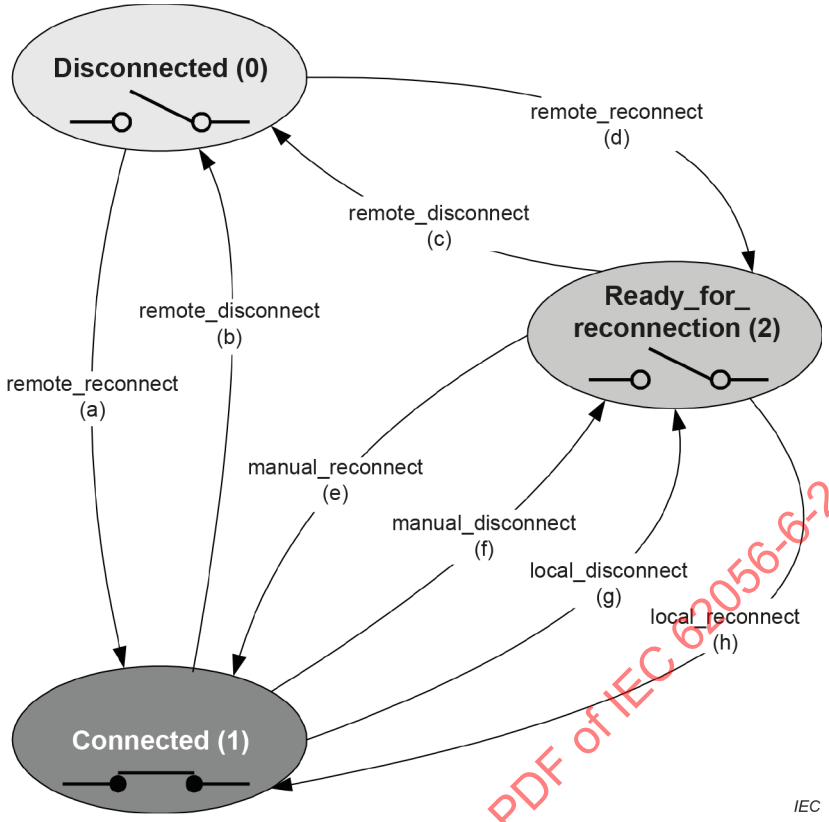


Figure 18 – State diagram of the Disconnect control IC

Disconnect and reconnect can be requested:

- Remotely, via a communication channel: remote_disconnect, remote_reconnect;
- Manually, using e.g. a push button: manual_disconnect, manual_reconnect;
- Locally, by a function of the meter, e.g. limiter, prepayment: local_disconnect, local_reconnect.

The states and state transitions of the Disconnect control IC are shown in Table 31. The possible state transitions depend on the control mode. The Disconnect control object doesn't feature a memory, i.e. any commands are executed immediately.

To define the behaviour of the disconnect control object for each trigger, the control mode shall be set.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 31 – Disconnect control IC – states and state transitions

States		
State number	**State name**	**State description**
0	Disconnected	The *output_state* is set to FALSE and the consumer is disconnected.
1	Connected	The *output_state* is set to TRUE and the consumer is connected.
2	Ready_for_reconnection	The *output_state* is set to FALSE and the consumer is disconnected.
State transitions		
Transition	**Transition name**	**State description**
a	remote_reconnect	Moves the “Disconnect control” object from the Disconnected (0) state directly to the Connected (1) state without manual intervention.
b	remote_disconnect	Moves the “Disconnect control” object from the Connected (1) state to the Disconnected (0) state.
c	remote_disconnect	Moves the “Disconnect control” object from the Ready_for_reconnection (2) state to the Disconnected (0) state.
d	remote_reconnect	Moves the “Disconnect control” object from the Disconnected (0) state to the Ready_for_reconnection (2) state. From this state, it is possible to move to the Connected (2) state via the manual_reconnect transition (e) or local_reconnect transition (h).
e	manual_reconnect	Moves the “Disconnect control” object from the Ready_for_connection (2) state to the Connected (1) state.
f	manual_disconnect	Moves the “Disconnect control” object from the Connected (1) state to the Ready_for_connection (2) state. From this state, it is possible to move back to the Connected (2) state via the manual_reconnect transition (e) or local_reconnect transition (h).
g	local_disconnect	Moves the “Disconnect control” object from the Connected (1) state to the Ready_for_connection (2) state. From this state, it is possible to move back to the Connected (2) state via the manual_reconnect transition (e) or local_reconnect transition (h). NOTE 1 Transitions f) and g) are essentially the same, but their trigger is different.
h	local_reconnect	Moves the “Disconnect control” object from the Ready_for_connection (2) state to the Connected (1) state NOTE 2 Transitions e) and h) are essentially the same, but their trigger is different.

Disconnect control		**0…n**	**class_id = 70, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name	(static)	octet-string				x
2. output_state	(dyn.)	boolean				x + 0x08
3. control_state	(dyn.)	enum				x + 0x10
4. control_mode	(static)	enum				x + 0x18
Specific methods		**m/o**				
1. remote_disconnect (data)		m				x + 0x20
2. remote_reconnect (data)		m				x + 0x28

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.8.2 Attribute description

4.5.8.2.1 logical_name

Identifies the “Disconnect control” object instance. See 6.2.22 and 6.2.46.

4.5.8.2.2 output_state

Shows the actual physical state of the device connection the supply.

boolean: TRUE = (BOOLEAN 1) Connected,
FALSE = (BOOLEAN 0) Disconnected

NOTE 1 In electricity metering, the supply is connected when the disconnector device is closed.

NOTE 2 In gas and water metering, the supply is connected when the valve is open.

4.5.8.2.3 control_state

Shows the internal state of the disconnect control object.

enum:

(0) Disconnected,
(1) Connected,
(2) Ready_for_reconnection

4.5.8.2.4 control_mode

Configures the behaviour of the disconnect control object for all triggers, i.e. the possible state **transitions**.

control_mode	Disconnection				Reconnection			
	Remote		Manual	Local	Remote		Manual	Local
enum:	(b)	(c)	(f)	(g)	(a)	(d)	(e)	(h)
(0)	–	–	–	–	–	–	–	–
(1)	x	x	x	x	–	x	x	–
(2)	x	x	x	x	x	–	x	–
(3)	x	x	–	x	–	x	x	–
(4)	x	x	–	x	x	–	x	–
(5)	x	x	x	x	–	x	x	x
(6)	x	x	–	X	–	x	x	x
In Mode (0) the disconnect control object is always in 'connected' state. Local disconnection is always possible unless the corresponding trigger is inhibited.								

4.5.8.3 Method description

4.5.8.3.1 remote_disconnect (data)

Forces the “Disconnect control” object into 'disconnected' state if remote disconnection is enabled (control mode > 0).

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.8.3.2 remote_reconnect (data)

Forces the “Disconnect control” object into the 'ready_for_reconnection' state if a direct remote reconnection is disabled (control_mode = 1, 3, 5, 6).

Forces the “Disconnect control” object into the 'connected' state if a direct remote reconnection is enabled (control_mode = 2, 4).

data::= integer (0)

4.5.9 Limiter (class_id = 71, version = 0)

4.5.9.1 Overview

Instances of the “Limiter” IC allow defining a set of actions that are executed when the value of a *value* attribute of a monitored object “Data”, “Register”, “Extended Register”, “Demand Register”, etc. crosses the threshold value for at least minimal duration time.

The threshold value can be normal or emergency threshold. The *emergency threshold* is activated via the *emergency_profile* defined by *emergency profile id*, *emergency activation time*, and *emergency duration*. The *emergency profile id* element is matched to an *emergency profile group id*: this mechanism enables the activation of the emergency threshold only for a specific emergency group.

Limiter		**0...n**	**class_id = 71, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name	(static)	octet-string				x
2. monitored_value	(static)	value_definition				x + 0x08
3. threshold_active	(dyn.)	threshold				x + 0x10
4. threshold_normal	(static)	threshold				x + 0x18
5. threshold_emergency	(static)	threshold				x + 0x20
6. min_over_threshold_duration	(static)	double-long-unsigned				x + 0x28
7. min_under_threshold_duration	(static)	double-long-unsigned				x + 0x30
8. emergency_profile	(static)	emergency_profile				x + 0x38
9. emergency_profile_group_id_list	(static)	array				x + 0x40
10. emergency_profile_active	(dyn.)	boolean				x + 0x48
11. actions	(static)	action				x + 0x50
Specific methods		***m/o***				

4.5.9.2 Attribute description

4.5.9.2.1 logical_name

Identifies the “Limiter” object instance. See 6.2.15.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.9.2.2 monitored_value

Defines an attribute of an object to be monitored. Only attributes with simple data types are allowed.

```
value_definition::= structure
{
    class_id:          long-unsigned,
    logical_name:      octet-string,
    attribute_index:   integer
}
```

4.5.9.2.3 threshold_active

Provides the active threshold value to which the attribute monitored is compared.

threshold: The threshold is of the same type as the attribute monitored.

4.5.9.2.4 threshold_normal

Provides the threshold value to which the attribute monitored is compared when in normal operation.

threshold: see 4.5.9.2.3.

4.5.9.2.5 threshold_emergency

Provides the threshold value to which the attribute monitored is compared when an emergency profile is active.

threshold: see 4.5.9.2.3.

4.5.9.2.6 min_over_threshold_duration

Defines minimal over threshold duration in seconds required to execute the over threshold action.

4.5.9.2.7 min_under_threshold_duration

Defines minimal under threshold duration in seconds required to execute the under threshold action.

4.5.9.2.8 emergency_profile

An *emergency_profile* is defined by three elements: emergency_profile_id, emergency_activation_time, emergency_duration.

An emergency profile is activated if the emergency_profile_id element matches one of the elements on the emergency_profile_group_id_list, and time matches the emergency_activation_time and emergency_duration element:

```
emergency_profile::= structure
{
    emergency_profile_id:          long-unsigned,
    emergency_activation_time:     octet-string,
    emergency_duration:            double-long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

– the emergency_activation_time element defines the date and time when the emergency_profile activated. The octet-string is encoded as specified in 4.1.6.1 for *date-time*,

– the emergency_duration element defines the duration in seconds, for which the *emergency_profile* is activated.

When an emergency profile is active, the *emergency_profile_active* attribute is set to TRUE.

4.5.9.2.9 emergency_profile_group_id_list

Defines a list of group id-s of the emergency profile.

The emergency profile can be activated only if emergency_profile_id element of the *emergency_profile* attribute matches one of the elements on the *emergency_profile_group_id_list*:

```
array        emergency_profile_group_id

emergency_profile_group_id::= long-unsigned
```

4.5.9.2.10 emergency_profile_active

Indicates that the emergency_profile is active.

4.5.9.2.11 actions

Defines the scripts to be executed when the monitored value crosses the threshold for minimal duration time.

```
action::= structure
{
        action_over_threshold:      action_item,
        action_under_threshold:     action_item
}
```

Where:

– action_over_threshold defines the action when the value of the attribute monitored crosses the threshold in upwards direction and remains over threshold for minimal over threshold duration time;

– action_under_threshold defines the action when the value of the attribute monitored crosses the threshold in the downwards direction and remains under threshold for minimal under threshold duration time.

```
action_item::= structure
{
        script_logical_name:    octet-string,
        script_selector:        long-unsigned
}
```

4.5.10 Parameter monitor (class_id = 65, version = 1)

4.5.10.1 General

Instances of the “Parameter monitor” IC are used to monitor a list of COSEM object attributes that hold parameters.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If the value of an attribute in the *parameter_list* changes, the identifier and the value of that attribute is automatically captured to the *changed_parameter* attribute. The time when the change of the parameter occurred is captured in the *capture_time* attribute. These attributes may be captured by a “Profile generic” object. In this way, a log of all parameter changes can be built. For the OBIS code of the Parameter monitor log objects, see IEC 62056-6-1:2023, 6.5.

NOTE 1 In the case of simultaneous or quasi simultaneous parameter changes the order of capturing and logging the changed parameters has to be managed by the application.

Several Parameter monitor objects and corresponding Profile generic objects can be instantiated to manage a number of parameter groups. The link between the Parameter monitor object and the corresponding Profile generic object is via the *capture_object* attribute of the Profile generic object.

NOTE 2 As the various parameters may be of different type and length, the entries in the profile column holding the parameters will be also of different type and length. This can be managed by capturing different kind of parameters into different Parameter list Profile generic objects and parameter logs.

NOTE 3 The Profile generic object holding the parameter change log may capture other suitable object attributes, such as the *time* attribute of the Clock object and any other relevant values.

– Use of the Parameter monitor IC for detecting configuration parameter changes:

Each client that has access to the server has to be aware of the current parametrisation in order to be able to correctly exchange data with servers. The parameters may be grouped and each group may have a name.

Although the parametrisation may be known initially, it may change during the lifetime of the meter, for example it may be changed by another client such as a field service device.

The current configuration may always be retrieved by reading the configuration parameters at the beginning of the exchange. This is not efficient and in the case of push operation it is not practical.

A solution is needed that allows clients to verify if the configuration of the server is as expected or to detect if any change has occurred.

To achieve this, version 1 of the Parameter monitor IC specifies new attributes to hold:

- the parameter_list_name;
- the hash_algorithm_id;
- the parameter_value_digest; and
- the parameter values.

When a client configures the server, it can retrieve these attributes and share them with other clients.

When a client starts a data exchange with the server, it can read the parameter_value_digest attribute and compare it to the value it has stored. If they match, the configuration is as expected, if not, the client can read the parameter_values attribute to identify which one(s) have changed.

The attributes can also be captured into Profile generic object buffer attributes to assure the validity of the data captured. The resulting overhead can be minimized by the use of null-data encoding.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Parameter monitor	0...n	class_id = 65, version = 1			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. changed_parameter	structure				x + 0x08
3. capture_time	date-time				x + 0x10
4. parameter_list	array				x + 0x18
5. parameter_list_name	octet-string				x + 0x20
6. hash_algorithm_id	enum				x + 0x28
7. parameter_value_digest	octet-string				x + 0x30
8. parameter_values	structure				x + 0x38
Specific methods	***m/o***				x + 0x40
1. add_parameter (data)	o				x + 0x48
2. delete_parameter (data)	o				x + 0x50

4.5.10.2 Attribute description

4.5.10.2.1 logical_name

Identifies the “Parameter monitor” object instance. See 6.2.14.

4.5.10.2.2 changed_parameter

Holds the identifier and the value of the most recently changed parameter.

```
structure
{
    class_id:        long-unsigned,
    logical_name:    octet-string,
    attribute_index: integer,
    attribute_value: CHOICE
    -- CHOICE as specified in the “Data” interface class
}
```

4.5.10.2.3 capture_time

Provides data and time information showing when the value of the *changed_parameter* attribute has been captured.

date-time is formatted as specified in 4.1.6.1.

4.5.10.2.4 parameter_list

Holds the list of parameters managed by a given instance of the “Parameter monitor” IC.

```
parameter_list::= array parameter_list_element

parameter_list_element::= structure
{
    class_id:        long-unsigned,
    logical_name:    octet-string,
    attribute_index: integer
}
```

NOTE The list of parameters monitored may be changed by using the *add_parameter* or *delete_parameter* methods or writing this attribute.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.10.2.5 parameter_list_name

Holds the name given to the list of parameter as defined by the parameter_list attribute.

4.5.10.2.6 hash_algorithm_id

enum

(0) SHA-256,
(1) SHA-384,
(2) Last 16 bytes of SHA-256,
(3) Last 8 bytes of SHA-256,
(4) Last 4 bytes of SHA-256

4.5.10.2.7 parameter_value_digest

Result of the digest calculation according to the algorithm specified by the hash_algorithm_id attribute. The digest is calculated on all the parameter list values as defined by attribute parameter_list in the order they are placed in the array. To calculate the digest, the whole set of parameter_list attribute values is first transformed into an octet-string. Rules for this transformation are defined IEC 62056-5-3:2023, 5.3.4.3. Data conversion, for all the data types which are not an octet-string.

4.5.10.2.8 parameter_values

Holds a copy of the values of the attributes referenced in parameter_list attribute.

It is a structure that holds the A-XDR encoded value of each attribute in the order of the parameter_list array.

4.5.10.3 Method description

4.5.10.3.1 add_parameter (data)

Adds one parameter to the *parameter_list*.

data::= parameter_list_element

NOTE A parameter can be logged as soon as it is added to the list. Adding an element to the list does not affect the *buffer* of the “Profile generic” object capturing the *changed_parameter* attribute.

4.5.10.3.2 delete_parameter (data)

Deletes one parameter from the *parameter_list*.

data::= parameter_list_element

NOTE When a parameter is deleted from the parameter list, its changes will not be logged any more. Removing an element from the list does not affect the *buffer* of the “Profile generic” object capturing the *changed_parameter* attribute.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11 Sensor manager (class_id = 67, version = 0)

4.5.11.1 General

NOTE This interface class is used mainly in metering energy types other than electricity.

Most measuring instruments under the scope of the MID operate with dedicated sensors (transducers and transmitters) connected to the processing unit. These sensors have to be permanently supervised concerning their functioning and limits to fulfil the metrological requirements for subsequent calculation of monetary values.

In addition, the measured values have to be monitored. These values may be related to a physical quantity – raw values of voltage, current, resistance, frequency, digital output – provided by the sensor, and the measured quantities resulting from the processing of the information provided by the sensor.

It is necessary to monitor and often to log the relevant values in order to obtain diagnostic information that allows:

- the identification of the sensor device;
- the connection and the sealing status of the sensor;
- the configuration of the sensors;
- the monitoring of the operation of the sensors;
- the monitoring of the result of the processing.

The “Sensor manager” interface class allows managing detailed information related to a sensor by a single object.

For simpler sensors / devices, already existing COSEM objects – identifying the sensors, holding measurement values and monitoring those measurement values – can be used.

4.5.11.2 Overview

Instances of the “Sensor manager” IC manage complex information related to sensors. They also allow monitoring the raw data and the processed value, derived by processing the raw-data using appropriate algorithms as required by the particular application. This IC includes a number of functions:

- nameplate data of the sensor and site information (attributes 2 to 6);
- an “Extended register” function for the *raw-value* (attributes 7 to 10);
- a “Register monitor” function for the *raw-value* (attributes 11-12);

  NOTE 1 Not every raw data (e.g. the voltage output of a pressure sensor) has its own OBIS code / object. This is the reason to include raw data in the Sensor manager class.

- a “Register monitor” function for the *processed_value* (attributes 13 to 15).

  NOTE 2 Not all “modules” are necessarily present. The attributes not used are possibly not implemented or not accessible.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Sensor manager		0...n	class_id = 67, version = 0			
Attribute (s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. serial_number	(dyn,)	octet-string				x + 0x08
3. metrological_identification	(dyn.)	octet-string				x + 0x10
4. output_type	(dyn.)	enum				x + 0x18
5. adjustment_method	(dyn.)	octet-string				x + 0x20
6. sealing_method	(dyn.)	enum				x + 0x28
7. raw_value	(dyn.)	CHOICE				x + 0x30
8. scaler_unit	(dyn.)	structure				x + 0x38
9. status	(dyn.)	CHOICE				x + 0x40
10. capture_time	(dyn.)	date-time				x + 0x48
11. raw_value_thresholds	(dyn.)	array				x + 0x50
12. raw_value_actions	(dyn.)	array				x + 0x58
13. processed_value	(dyn.)	processed_value_ definition				x + 0x60
14. processed_value_thresholds	(dyn.)	array				x + 0x68
15. processed_value_actions	(dyn.)	array				x + 0x70
Specific methods		***m/o***				
1. reset (data)		*o*				x + 0x80

4.5.11.3 Attribute description

4.5.11.3.1 logical_name

Identifies the “Sensor manager” object instance.

NOTE Sensor manager objects are used in relation to non-electricity metering. Therefore no OBIS codes are defined in this document.

4.5.11.3.2 serial_number

Identifies the sensor (->site information).

NOTE For simple sensors, the serial number may be held by “Data” objects with appropriate OBIS codes if this is the only information required.

4.5.11.3.3 metrological_identification

Describes metrologically relevant information of the sensor, e.g. metrological identifier, calibration date.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11.3.4 output_type

describes the physical output of the sensor (->site information)

enum:

(0) not specified,
(1) 4–20 mA,
(2) 0–20 mA
(3) 0–5 V,
(4) 0–10 V,
(5) Pt100,
(6) Pt500,
(7) Pt1000,
(8) HART
(128) manufacturer specific
All other values are reserved.

4.5.11.3.5 adjustment_method

Describes the sensor adjustment method, e.g. by 3-Measuring-point-equation.

4.5.11.3.6 sealing_method

Type of seals applied to the sensor.

enum:

(0) none,
(1) mechanical (e.g. wire or protective sticker);
(2) electronic (e.g. contact);
(3) software (e.g. password protection)

4.5.11.3.7 raw_value

Physical value from the sensor (e.g. voltage from a pressure to voltage converter).

For the possible data type choices, see the “Register” IC in 4.3.3.2.2.

4.5.11.3.8 scaler_unit

The scaler and the unit of the *raw_value*.

For the definition of *scaler_unit,* see the “Register” IC in 4.3.3.2.3.

4.5.11.3.9 status

Status of last *raw_value* captured (for the definition of status, see the “Extended register” IC, 4.3.3.2.4). The status keeps information such as:

– sensor failure,
– sensor activated / inactivated.

The meaning of the elements of the *status* shall be provided for each “Sensor manager” object instance.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11.3.10 capture_time

Provides a “Sensor manager” object specific date and time information showing when the value of the attribute *raw_value* has been captured.

date_time is formatted as specified in 4.1.6.1.

For the possible data type choices, see the “Extended register” interface class in 4.3.3.2.5.

4.5.11.3.11 raw_value_thresholds

Provides the threshold values to which the *raw_value* attribute is compared.

array threshold

threshold: The threshold is of the same type as the raw-value (4.5.11.3.7).

4.5.11.3.12 raw_value_actions

Defines the scripts to be executed when the *raw_value* crosses the corresponding threshold. The attribute *raw_value_actions* has exactly the same number of elements as the attribute *raw_value_thresholds*. The ordering of the *action_items* corresponds to the ordering of the thresholds. For the specification of actions, see the *Register monitor* IC in 4.5.6.2.4.

4.5.11.3.13 processed_value

References the attribute holding the processed value of the raw data provided by the sensor.

```
processed_value_definition:= structure
{
        class_id:           long-unsigned,
        logical_name:       octet-string,
        attribute_index:    integer
}
```

Note that more than one “Sensor manager” objects and processed value objects may belong to the same sensor.

4.5.11.3.14 processed_value_thresholds

Provides the threshold values to which the processed value is compared.

array threshold

threshold: The threshold is of the same type as the value processed. (referenced by the *processed-value* attribute 4.5.11.3.13).

4.5.11.3.15 processed_value_actions

Defines the scripts to be executed when the processed value crosses the corresponding threshold.

The attribute *processed_value_actions* has exactly the same number of elements as the attribute *processed_value_thresholds*. The ordering of the *action_items* corresponds to the ordering of the thresholds.

For the specification of actions, see the *Register monitor* IC in 4.5.6.2.4.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11.4 Method description

4.5.11.4.1 reset (data)

This method resets the *raw_value* to the default value. The default value is an instance specific constant.

The attribute *capture_time* is set to the time of the reset execution.

data::= integer (0)

4.5.11.5 Example for absolute pressure sensor

Figure 19 illustrates the definition of relevant upper and lower thresholds.

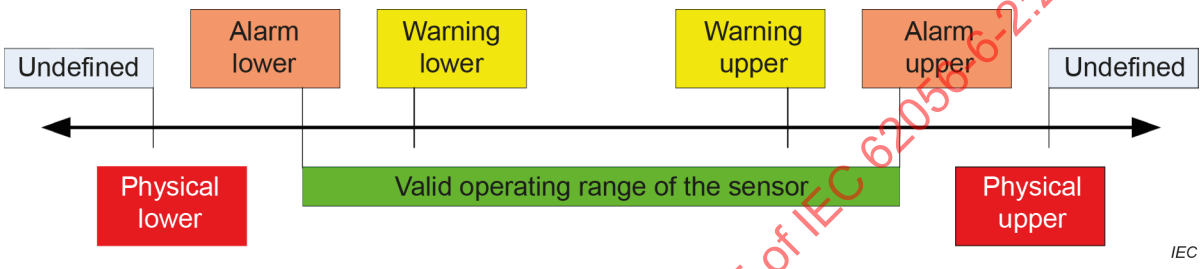


Figure 19 – Definition of upper and lower thresholds

Table 32 and Table 33 show examples of the various thresholds and the actions performed when the thresholds are crossed.

Table 32 – Explicit presentation of threshold value arrays

Threshold	Physical lower	Physical upper	Alarm lower	Alarm upper	Warning lower	Warning upper
Value	1,0	5,5	1,2	5,0	1,4	4,8
scaler_unit	1, Volt	1, Volt	1, bar	1, bar	1, bar	1, bar

Table 33 – Explicit presentation of action_sets

action_set	Physical lower	Physical upper	Alarm lower	Alarm upper	Warning lower	Warning upper
action_up	clr_phy_ alarm_bit	set_phy_ alarm_bit	clear_ alarm_bit	set_ alarm_bit	clear_ warn_bit	set_ warn_bit
action_down	set_phy_ alarm_bit	clr_phy_ alarm_bit	set_ alarm_bit	clear_ alarm_bit	set_ warn_bit	clear_ warn_bit

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.12 Arbitrator (class_id = 68, version = 0)

4.5.12.1 Overview

Instances of the “Arbitrator” IC allow determining, based on pre-configured rules comprising permissions and weightings, which action is carried out when multiple actors may request potentially conflicting actions to control the same resource. Generally, there is one “Arbitrator” object instantiated for each resource for which competing action requests need to be handled.

The “Arbitrator” IC allows:

- configuring the possible, potentially conflicting actions that can be requested;
- configuring the permissions for each actor to request the possible actions;
- configuring the weighting for each actor for each possible request.

NOTE 1 Examples for a resource are the supply control switch or a gas valve of the meter. Examples for possible actions are disconnect supply, enable reconnection, reconnect supply, prevent disconnection, prevent reconnection.

The actions that can be requested are held in the *actions* attribute as an array of script identifiers. The scripts – held by separate “Script table” objects – allow performing a wide range of functions. There may be actions designed to inhibit the execution of other actions: these are modelled as null-scripts, i.e. pointing to a script_identifier (0) of a “Script table” object.

NOTE 2 The “Arbitrator” objects do not contain the names of the actions, but their purpose and effect can be deduced by looking at the relevant “Script table” objects.

The permissions are held in the *permissions_table* attribute as an array of bit-strings: each element in the array represents one actor and each bit in the bit-string represents one action from the *actions* array.

The weightings are held in the *weightings_table* attribute as a two-dimensional array: each line represents one actor and there is one weight allocated to each possible action for that actor. For actions designed to inhibit other actions a very high weight may be allocated compared to the weight of the action that is to be inhibited.

Actions are requested by invoking the *request_action* method. The method invocation parameters contain:

- the identifier of the actor. This element, an unsigned number, points to a line in the *permissions_table, weightings_table* and *most_recent_requests_table* attributes;

  NOTE 3 Names of the actors may be specified in project specific companion specifications.

- the list of actions requested, in the form of a bit-string. Each bit corresponds to one element of the *actions* array: for the actions requested the bit is set to 1, for the actions not requested (inactions) the bit is set to 0. An actor may request none, one, several or all actions in a single request in one invocation. The reason to allow requesting multiple actions in a single request is to allow the actor to request an action, and at the same time to prevent another actor reversing that action.

  NOTE 4 For example, an actor may request disconnecting the supply and preventing another actor to reconnect it.

  NOTE 5 An earlier action request by an actor can be cleared by not requesting the same action (i.e. by requesting an inaction) in another invocation of the *request_action* method by the same actor. With this, a request for inhibiting an action is lifted.

The *most_recent_requests_table* attribute holds the list of the most recent request of each actor, in the form of an array of bit-strings: each element in the array represents the last request of an actor, and each bit in the bit-string represents one action / inaction requested.

When the *request_action* method is invoked by an actor the AP carries out the following activities:

- it checks the *permissions_table* attribute entry for the given actor to ascertain if the actions requested are permitted or not;
- it updates the *most_recent_requests_table* attribute by setting or clearing the bit in the bit-string for that actor for each action requested that is also permitted (bit is set); or not requested / not permitted (bit is cleared);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- it applies the *weightings_table* for the *most_recent_requests_table:* for each bit set in the *most_recent_requests_table* the corresponding weight of each actor is applied;
- for each action, the weights are summed; and then
- if there is a unique highest total weight for an action, this value is written to the *last_outcome* attribute and the corresponding script is executed. If there is no highest unique total weight, nothing happens.

Arbitrator		**0...n**	**class_id = 68, version = 0**			
Attribute description		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. actions	(static)	array			all empty	x + 0x08
3. permissions_table	(static)	array			all bits cleared	x + 0x10
4. weightings_table	(static)	array			all zero	x + 0x18
5. most_recent_requests_table	(dyn.)	array			all bits cleared	x + 0x20
6. last_outcome	(dyn.)	unsigned	0	n	0	x + 0x28
Specific methods		**m/o**				
1. request_action (data)		m				x + 0x30
2. reset (data)		o				x + 0x38

4.5.12.2 Attribute description

4.5.12.2.1 logical_name

Identifies the “Arbitrator” object instance. See 6.2.47.

4.5.12.2.2 actions

Defines the actions that can be requested.

```
actions::= array action_item

action_item::= structure
{
    script_logical_name:    octet-string,
    script_selector:        long-unsigned
}
```

Where:

– script_logical_name: defines the *logical_name* of the “Script table” object;

– script_selector: defines the script_identifier of the script to be executed.

Entries that are intended to inhibit other actions are represented by null-scripts, i.e. pointing to script_identifier (0) of a “Script table” object.

4.5.12.2.3 permissions_table

Contains the permissions for each actor to request actions.

```
permissions_table::=    array       actor_permissions

actor_permissions::=    bit-string
```

Each entry represents the permissions for one actor to request each action.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Each bit in the bit-string corresponds to one action in the *actions* array. The length of the bit-string shall be the same as the number of elements in the *actions* array.

The leading bit corresponds to the first element and the trailing bit corresponds to the last element in the *actions* array.

The bits shall be set for actions allowed and cleared for actions not allowed.

NOTE These *actor_permissions* model business rules rather than acting as a form of access control.

4.5.12.2.4 weightings_table

Holds the weight allocated for each actor and to each possible action of that actor.

```
weightings_table::=      array actor_weighting_list

actor_weighting_list::=  array actor_action_weight

actor_action_weight::= long-unsigned
```

The number and order of elements in the *weightings_table* array shall be the same as in the *permissions_table* array.

The number of elements in the actor_weighting_list array shall be the same as in the *actions* array. The first element shows the weight for the first action and the last element shows the weight of the last action.

NOTE It is preferred using powers of 2 as weights. Using different weights for different actors and actions helps to avoid situations when there is no unique outcome after evaluating the action request (in which case no action is performed).

4.5.12.2.5 most_recent_requests_table

Holds the most recent requests of each actor.

```
most_recent_requests_table::= array most_recent_request
most_recent_request::=        bit-string
```

Each entry represents the most recent request of an actor.

The number and order of elements in the *most_recent_requests_table* array shall be the same as in the *permissions_table* array.

The length of the most_recent_request bit-string shall be the same as the number of elements in the *actions* array. The leading bit corresponds to the first element and the trailing bit corresponds to the last element in the *actions* array.

For each action that has been requested and which is also permitted the bit is set. For each action that is not requested (inaction) or not allowed the bit is cleared.

4.5.12.2.6 last_outcome

Contains the outcome of the most recent request that has resulted a unique highest total weight for an action requested and therefore performed.

The number identifies a bit in the request_action_list bit-string: the number 1 corresponds to the leading bit and consequently to the first element in the *actions* array.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.12.3 Method description

4.5.12.3.1 request_action (data)

Defines the actions that are requested by an actor.

```
data::= structure
{
        request_actor:          unsigned,
        request_action_list:    bit-string
}
```

Where:

– request_actor is an index into the corresponding arrays. The number 1 identifies the first entry in the *permissions_table*, the first entry of the *weightings_table* and the first entry in the *most_recent_requests_table* arrays. The number *n* identifies the highest entry in these arrays;

– request_action_list identifies the action(s) to be performed. For each action requested the bit is set. For each action not requested (inaction) the bit is not set. The length of the bit-string shall be the same as the number of elements in the *actions* array. The leading bit corresponds to the first element and the trailing bit corresponds to the last element.

Although several actions can be requested, only one action (modelled by a script) will be actually performed, provided that it is permitted for that actor and there is a single highest total outcome. As specified above, an action may be a null-script.

4.5.12.3.2 reset (data)

Clears the configurable fields of the object instance, that is:

– clears all bits in the *permissions_table* attribute;

– sets all values in the *weightings_table* attribute to zero;

– clears all bits in the *most_recent_requests_table* attribute;

– sets the *last_outcome* attribute to zero.

NOTE Following a reset it can be expected that every invocation of *request_action* will leave the *last_outcome* attribute equal to zero, and no action will be performed since the elements in the *permissions_table* attribute are all cleared.

```
data::= integer (0)
```

4.5.13 Modelling examples: tariffication and billing

Figure 20 shows an example of modelling tariff parametrization and management using COSEM objects.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

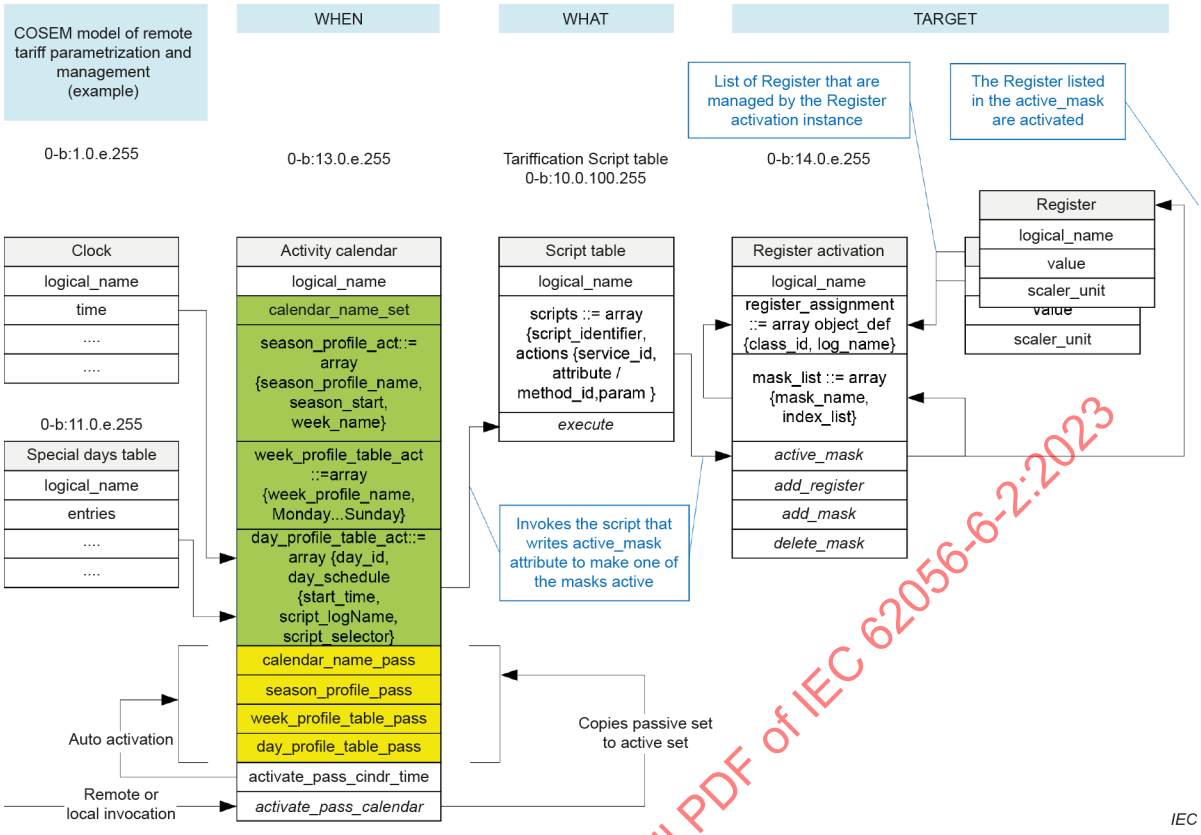


Figure 20 – COSEM tariffication model (example)

Figure 21 shows an example of modelling parametrization and management of billing using COSEM objects.

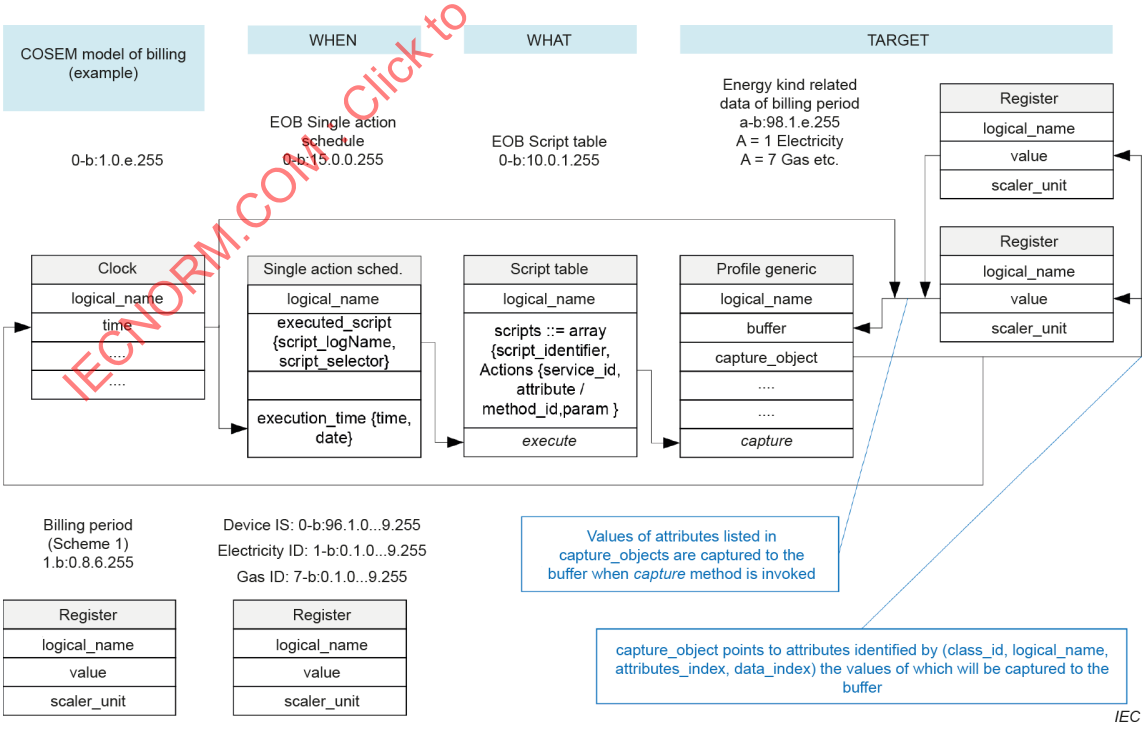


Figure 21 – COSEM billing model (example)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6 Payment metering related interface classes

4.6.1 Overview of the COSEM accounting model

The COSEM accounting model contains four interlinked interface classes: “Account”, “Credit”, “Charge” and the “Token gateway” IC. These classes are concerned with accounting for energy, not with delivery of that energy. The “Account” is linked to it’s associated “Credit”, “Charge” and “Token gateway” objects by use of the value group D and B field such that an “Account” with D=0 should be linked to a “Token gateway” with D=40 and have a “Credit” objects with D=10 and “Charge” objects with D=20. Whereas an “Account” with D=1 should have “token gateway” with D=41, “Credit” objects with D=11 and “Charge” objects with D=21, etc. Multiple “Token gateway”, “Credit” and “Charge” objects related to the same “Account” are identified using different values in the value group E field.

An “Account” object contains summary information and coordinates information pertaining to Credits and Charges. There is a single “Account” object per supply, for example, electricity import has one “Account” object, but a system that also has micro-generation could have a second “Account” to deal with the export of generated electricity; the second “Account” might or might not be accessible via the same Application Association (AA) as the first.

A “Credit” object contains detailed information about one source of funds. There is one or (usually) more “Credit” object(s) associated with an “Account”; for example, one object for token credit and one object for emergency credit. Both of these objects can receive credit amounts from tokens, but emergency credit can only receive credit amounts when it has been consumed (entirely or partially) and when *credit_configuration* has bit 2 (Requires the credit amount to be paid back) set.

There are several types of credit listed in IEC TR 62055-21:2005, and these are the types supported by the ”Credit” IC. There can be zero or more instances of each type of Credit.

- **token_credit:** Credit that is transferred to a meter operating in prepayment mode, normally in the form of Credit Tokens;

  NOTE 1 In a meter operating in credit mode or managed payment mode, a “Credit” object configured with type token_credit is used for recording the amount of credit used since last synchronised with a client.

  NOTE 2 The content of the token is not defined by this document and may be an amount of money or another quantity that can be accounted in a way that is equivalent to the currency used by the meter.

- **reserved_credit:** Credit that is held in reserve, which is released under specific conditions;
- **emergency_credit:** Accounting functions that deal with the calculation and transacting of credit that is released only under emergency situations. Usually the amount of emergency credit used is recovered from subsequently purchased credit token;

  NOTE 3 Emergency above refers to a time when a consumer does not have any token credit, and not to any safety related situation.

- **time_based_credit:** Credit that is released on a scheduled time basis;
- **consumption_based_credit:** Credit that is released on the basis of a schedule of consumption levels. For example if a consumer keeps their consumption below a threshold, the system may release a predefined amount of credit.

A “Charge” object contains detailed information about one sink of expenditure, that is, one way in which credit is being used up. There can be one or (usually) more “Charge” objects associated with an “Account” for instance, one for energy usage, one for standing charge, and possibly one paying off a debt such as an installation charge.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

There are several types of charge listed in IEC TR 62055-21:2005, but the following types, distinguished by the trigger for collection, are the ones most useful in the COSEM accounting model. There can be zero or more instances of each type of ”Charge” IC.

- **consumption_based_collection:** describes charges that are collected according to the amount of consumption that has occurred in a tariff. A price per unit is assigned to each tariff register of the energy consumed;

  NOTE 4 Tariffs cannot be applied when currency is in time or energy units.

- **time_based_collection:** describes charges that are collected regularly according to the passage of time, independent of consumption in that period. This may be used to collect standing charges, or debt charge to be paid off over a period of time;

- **payment_event_based_collection:** describes charges that are collected from every top-up that is received, typically for debt repayment. These may be expressed as **amount-based**, where a fixed amount is taken from each top-up credit received (for example, the consumer pays £2 out of every vend regardless of the vend amount), or ***percentage_based_collection*** where a proportion of the amount of top-up credit received is taken (for example, with every vend the consumer pays 20 % of the vend amount). Bit 0 (Percentage based collection) of the *charge_configuration* attribute of the “Charge” object specifies the method of *event_based_collection*. Figure 22 gives a general view of the account model.

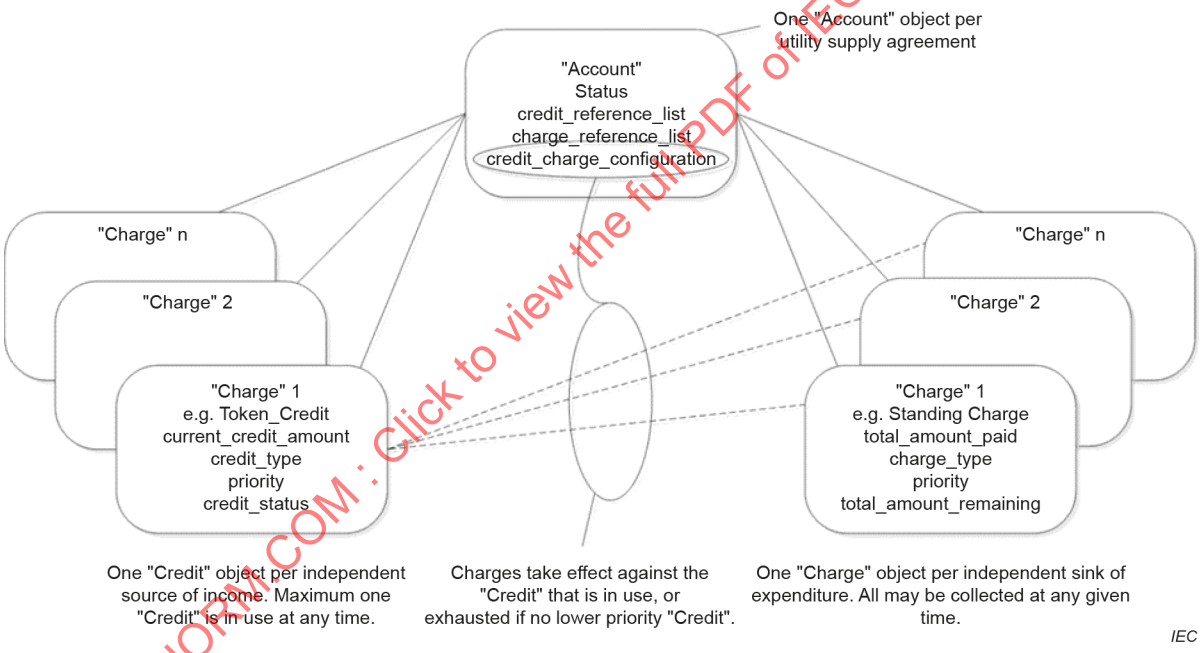


Figure 22 – Outline Account model

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 23 shows instances of “Account”, “Credit” and “Charge” interface classes with some of their attributes and the relationships between those attributes. In this example:

a) There is one “Account”, two “Credit” and two “Charge” objects configured;

b) “Credit” 1 is of type *token_credit* and the *low_credit_threshold* and *limit* attributes are configured to be 0;

c) nteraction between multiple classes is covered in this diagram. Detailed configuration of individual “Credit” and “Charge” objects is not shown.

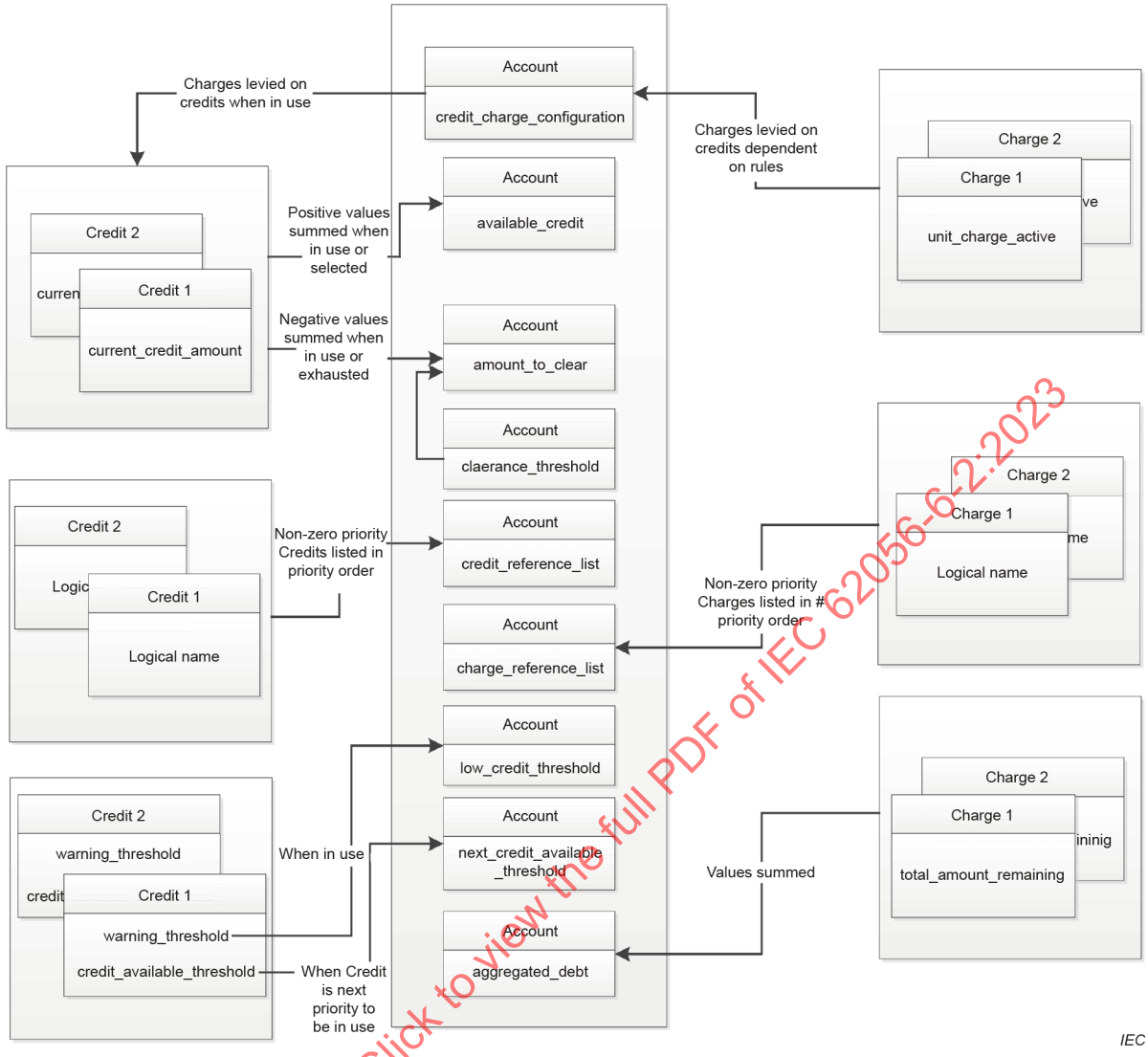


Figure 23 – Diagram of attribute relationships

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2 Account (class_id = 111, version = 0)

4.6.2.1 Overview

Instances of the “Account” IC manage all the necessary elements related to the supervision of the “Credit” objects and the “Charge” objects referenced by a particular instance of the “Account” IC.

The operation of the payment metering function will be defined within the configuration of “Charge”, “Credit”, and “Account” objects and disconnection rules.

NOTE Disconnection rules are either application specific or can be modelled using an instance of the “Arbitrator” IC, see 4.5.12.

If explicitly specified for a particular project, it is permissible for accounting to switch from credit to prepayment mode, or from prepayment to credit mode, once an accounting configuration has been correctly set up and the “Account” object has been activated. In the absence of any such specification the operating mode should remain fixed for any particular “Account” once it has been made active.

Account		0...n	class_id = 111, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string			n/a	x
2. account_mode_and_status		structure			0, 0	x + 0x08
3. current_credit_in_use	(dyn.)	unsigned			0	x + 0x10
4. current_credit_status	(dyn.)	bit-string			All bits clear	x + 0x18
5. available_credit	(dyn.)	double-long			0	x + 0x20
6. amount_to_clear	(dyn.)	double-long			0	x + 0x28
7. clearance_threshold	(static)	double-long			0	x + 0x30
8. aggregated_debt	(dyn.)	double-long			0	x + 0x38
9. credit_reference_list	(static)	array			empty	x + 0x40
10. charge_reference_list	(static)	array			empty	x + 0x48
11. credit_charge_configuration	(static)	array			empty	x + 0x50
12. token_gateway_configuration	(static)	array			empty	x + 0x58
13. account_activation_time	(static)	octet-string			n/a	x + 0x60
14. account_closure_time	(static)	octet-string			n/a	x + 0x68
15. currency	(static)	structure			n/a	x + 0x70
16. low_credit_threshold	(dyn.)	double-long			0	x + 0x78
17. next_credit_available_threshold	(dyn.)	double-long			0	x + 0x80
18. max_provision	(static)	long-unsigned			0	x + 0x88
19. max_provision_period	(static)	double-long			0	x + 0x90
Specific methods		**m/o**				
1. activate_account (data)		o				x + 0x98
2. close_account (data)		o				x + 0xA0
3. reset_account (data)		o				x + 0xA8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2 Attribute description

4.6.2.2.1 logical_name

Identifies the “Account” object instance. See 6.2.17.

4.6.2.2.2 account_mode_and_status

Defines the payment_mode, enumerated below, and the status of the “Account”, also enumerated.

```
account_mode_and_status::= structure
{
payment_mode: enum:
                    (1) Credit mode,
                    (2) Prepayment mode
account_status: enum:
                    (1) New (inactive) account,
                    (2) Account active,
                    (3) Account closed
}
```

The payment_mode is an indication of a notional prepayment or credit/managed payment mode.

NOTE Payment_mode does not force some other actions in the meter, or cause a change of behaviour.

The “Credit” and “Charge” objects associated with an “Account” object do not operate unless the “Account” object is in the active state. A New (inactive) “Account” object is passive and will become active at *account_activation_time* or invocation of the *activate_account* method.

4.6.2.2.3 current_credit_in_use

This attribute is an index into the *credit_reference_list* indicating which “Credit” object is *In use*.

4.6.2.2.4 current_credit_status

This attribute provides the status of the current “Credit” object *In use* and some information about the next priority “Credit” object.

Bit 0 = in_credit, set when the *available_credit* is above zero,

Bit 1= low_credit, set when the “Account” object’s *available_credit* level has passed below the “Account” object *low_credit_threshold*,

NOTE 1 The *low_credit_theshold* attribute of the associated “Credit” object *In use* is echoed in that attribute.

Bit 2 = next_credit_enabled, set when the “Account” object *credit_reference_list* contains at least one other “Credit” object with non-zero priority, regardless of credit level,

Bit 3 = next_credit_selectable, set when the next item in the “Account” object *credit_reference_list* contains a lower priority “Credit” object with non-zero priority with a *credit_configuration* bit 1 (Requires confirmation) set such that it requires confirmation (for example “Emergency Credit” that is selectable but has not yet been selected),

NOTE 2 The next “Credit” object becomes selectable when the immediately higher priority “Credit” object has reached the “Account” object’s *next_credit_available_threshold*.

Bit 4 = next_credit_selected, set when the ”Account” object *credit_reference_list* contains a “Credit” object of lower priority than the current “Credit” object *In use*, and that lower priority credit object is in the *Selected/Invoked* state,

NOTE 3 This is, for example, an “Emergency Credit” which has been selected but is not yet *In use*.

Bit 5 = selectable_credit_in_use, set when the previously selected credit in the “Account” object *credit_reference_list* is now *In use*,

Bit 6 = out_of_credit, set when the “Credit” object that was *In use* has been exhausted and no further credit is available without an action on the part of the consumer or other actor.

Bit 7 = RESERVED

When the next priority ”Credit” object becomes the current “Credit” object then all bits have to update.

NOTE 4 A “Credit” object becomes *Selectable* when it is the next priority “Credit” and the *next_credit_available_threshold* attribute (reflecting the *credit_available_threshold* attribute in the “Credit”) of the “Account” object is greater than or equal to the *available_credit* in the “Account” object. However if the *available_credit* becomes greater than the *next_credit_available_threshold* due to the selection of the “Credit” the status of the “Credit” remains (2) *Selected*. If the *available_credit* becomes greater than the *next_credit_available_threshold* due to a top up or method invocation to increase the *current_credit_amount* of a “Credit” objects that is (2) *Selected* or (3) *In use* then this will change the status of the “Credit” object to (0) *Enabled*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2.5 available_credit

The *available_credit* attribute is the sum of the positive *current_credit_amount* values in the instances of the “Credit” class that:

– are listed in the “Account” object *credit_reference_list;*

– and have a “Credit” object *credit_status* (2) *Selected/Invoked* or (3) *In use*.

This attribute holds the aggregate amount of credit available associated with the “Account” object. This value is positive or zero.

NOTE Negative values of “Credit“ object *current_credit_amount* attributes are summed in *amount_to_clear*. See also the vertical axis in Figure 25.

double-long, scaled according to the *currency* attribute.

4.6.2.2.6 amount_to_clear

This value is the sum of:

– all negative values of *current_credit_amount* in “Credit” objects that:
  - are listed in the “Account” object *credit_reference_list,* and
  - have a “Credit” object *credit_status* (4) *Exhausted*,
  - have the *credit_configuration* bit 2 (Requires the credit amount to be paid back) is cleared,

– the negative (Value * -1) of the amount of credit used from all “Credit” objects where the *credit_configuration* bit 2 (Requires the credit amount to be paid back) is set; and

– the negative (Value * -1) of the value of the “Account” object *clearance_threshold* attribute;

See also Figure 25.

NOTE 1 ”Credit” objects where the *credit_configuration* bit 2 is set are referred to as “repayable” credits.

Credit used is the difference between the *preset_credit_amount* and the *current_credit_amount* (thus a positive value) of a “Credit” object when the “Credit” is (3) *In use* or (4*) Exhausted*. In the case of non-repayable credits the credit used is not relevant.

NOTE 2 The payment meter’s application process might have specific functional behaviour to allow a consumer to live in emergency credit or not. This functionality is not within the scope of this Companion Specification.

This attribute is significant when the meter has accumulated a temporary debt, for instance, while an emergency credit is *In use*, and/or during a friendly credit period.

This attribute contains the minimum credit that the meter will need to receive (via token receipts and method invocations on “Credit” objects) in order to clear negative credits and also make “Credits” marked as repayable (for example an emergency credit), enabled again after they have been in the (3) *In use*, (2) *Selected/Invoked* or (4) *Exhausted* state.

NOTE 3 For an explanation of the process of distributing top ups between “Credit” objects, refer to attribute 12 *token_gateway_configuration*.

NOTE 4 Where a meter is being used in prepayment mode, *amount_to_clear* is usually the amount needed to reverse a disconnection. Amount to clear is shown on the vertical axis in Figure 25 as a negative value.

double-long, scaled according to the *currency* attribute (4.6.2.2.15).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2.7 clearance_threshold

This attribute is used in conjunction with the *amount_to_clear*, and is included in the description of that attribute.

This attribute becomes relevant when a meter has consumed all credit sources and has "Credit" objects with *current_credit_amount* attributes that have values less than zero.

This represents the value of credit that the *available_credit* attribute shall reach in order to make repayable "Credits" enabled again.

To achieve this, it is clear that enough credit has to be added to the meter to ensure that the current_credit_available attribute on all listed "Credit" objects is zero or greater.

double-long, scaled according to the currency attribute.

4.6.2.2.8 aggregated_debt

This attribute is a simple sum of *total_amount_remaining of* all the "Charge" objects which are listed in the "Account" object *charge_reference_list,* where bit 1 (Continuous collection) of the *charge_configuration* is cleared.

NOTE This value is not interchangeable with *amount_to_clear*; it is provided to assist external accounting.

double-long, scaled according to the *currency* attribute.

4.6.2.2.9 credit_reference_list

This attribute is an array of logical names, identifying a collection of "Credit" objects that operate with this "Account" object. The elements in the array shall appear in priority order (priority 1 being first to priority n being last).

Priority 0 credits shall not appear in this list, as by definition they are not enabled.

credit_reference_list::= array credit_reference

credit_reference::= octet-string

4.6.2.2.10 charge_reference_list

This attribute is an array of logical names, identifying a set of "Charge" objects that operate with this "Account" object. The elements in the array shall appear in priority order (priority 1 being first to priority n being last). Priority 0 charges shall not appear in this list, as by definition they are not applied.

charge_reference_list::= array charge_reference

charge_reference::= octet-string

4.6.2.2.11 credit_charge_configuration

This attribute maps out which Charges are to be collected from which Credits.

If the array has zero elements then it is assumed that any Charge may be collected from any Credit in accordance with the "Credit" objects *credit_status being (3) In use or (4) Exhausted.*

If there are entries in this array then they represent each Charge that can be collected from each Credit.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If there is a Credit from which no Charges are collected then that Credit is not consumed and will remain unchanged until a Charge is configured.

NOTE 1 Collection of a Charge will cease when reaching zero, in cases where the appropriate "Charge" object has bit 1 (continuous collection) of its *charge_configuration* cleared. This will not alter the "Account" *credit_charge_configuration.*

```
credit_charge_configuration::= array credit_charge_configuration_element

credit_charge_configuration_element::= structure
{
    credit_reference:          octet-string,
    charge_reference:          octet-string,
    collection_configuration:  bit-string
}
```

Where *credit_reference* and *charge_reference* contain the *logical_name* of the relevant "Credit" and "Charge" object.

```
collection_configuration::= bit-string
```

This element defines behaviour under specific conditions.

Bit 0 = Collect when supply disconnected. When set, the "Charge" referred to in *charge_reference* may be collected when supply is disconnected. When cleared, the "Charge" referred to in *charge_reference* shall not be collected when supply is disconnected.

Bit 1 = Collect in load limiting periods. When set, the "Charge" referred to in *charge_reference* may be collected when supply is in a load limiting period. When cleared, the "Charge" referred to in *charge_reference* shall not be collected when supply is in a load limiting period.

Bit 2 = Collect in friendly credit periods. When set, the "Charge" object referred to in *charge_reference* may be collected when supply is in a friendly credit period. When cleared, the "Charge" referred to in *charge_reference* shall not be collected when supply is in a friendly credit period.

NOTE 2 The meter application knows internally whether it is in a supply disconnected, load limiting period, or friendly credit period, and takes appropriate action based on the value of this attribute.

NOTE 3 It is implicit that charges are also collected at times when these specific conditions are not present.

4.6.2.2.12 token_gateway_configuration

This attribute is designed to configure how a new top-up token from the "Token gateway" is to be apportioned, such that a configurable percentage of the token amount is distributed to each "Credit" object.

NOTE 1 The distribution of token top up amounts is also affected by the values of the *current_credit_amount*, *credit_type* and *credit_status* attributes of the "Credit" object.

If there are restrictions on how the credit token received through the "Token gateway" object is distributed, then this attribute determines the minimum proportion of the credit token that is attributed to each "Credit" object referenced in the array.

The proportions in this array shall not add up to more than 100 %. If the sum of the proportions is less than 100 % then the remainder will be added to the "Credit" objects using the rules below for an empty *"token_gateway_configuration"* attribute.

NOTE 2 It is possible that some "Credits" will get more than their allocated proportion as a result of the process highlighted above. The credit token is always 100 % distributed across the "Credit" objects.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE 3 The connection between a meter's one or more "Token gateway" objects and a specific "Account" object is not explicitly shown in the model. The management of multiple "Account" objects and multiple token gateways is the subject of project specific companion specifications. However in the normal case an "Account" object has one "Token gateway" object and all tokens received follow the rules associated with the "Account" object with which it is associated (by application of the value group D field; see also 4.6.1).

```
token_gateway_configuration::= array
                                token_gateway_configuration_element

token_gateway_configuration_element::= structure
{
    credit_reference        octet-string,
    token_proportion        unsigned;
}
```

Where:

– credit_reference contains the *logical_name* of the relevant "Credit" object;

– token_proportion is scaled as one percent per integer step from 0 to 100.

If there are no entries in the array then it is assumed that there are no restrictions on processing the credit token within the meter's payment application and credit should be applied first to the lowest priority "Credit" object with its *credit_status* set to *(3) In use or* (4) *Exhausted,* then apportioned to the other "Credit" objects in ascending order of the "Credit" object *priority* (starting with priority n and ending with priority 1).

If the "Credit" object requires a limited amount of top up (for example, an Emergency "Credit" object that is being repaid in full) then the credit used (*preset_credit_amount* less *current_credit_amount* before the top up) is taken from the Token amount and any surplus is applied to higher priority *Credit*s following the same rules. At the point when the credit used is repaid the "Credit" will move from (3) *In use* or (4) *Exhausted* to (0) *Enabled.*

See also Note 1.

4.6.2.2.13 account_activation_time

Defines the time when the object itself invokes the specific method *activate_account*. A definition with "not specified" notation in all fields of the attribute will not be acted upon. Partial "not specified" notation in some fields of date and time are not allowed.

When this time is in the past, this field indicates the time at which the "Account" object was activated.

octet-string, formatted as specified in 4.1.6.1 for *date_time.*

4.6.2.2.14 account_closure_time

Defines the time when the object itself invokes the specific method *close_account*. A definition with "not specified" notation in all fields of the attribute will not be acted upon. Partial "not specified" notation in just some fields of date and time are not allowed.

When this time is in the past, this field indicates the time at which the "Account" object was closed.

octet-string, formatted as specified in 4.1.6.1 for *date_time.*

4.6.2.2.15 currency

Defines the "currency" unit used by all functions of an "Account" object. The *charge_per_unit* in the *unit_charge* attribute of "Charge" objects has an additional scaling factor that can be applied.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The units could be currency or other units such as kWh, minutes; or other consumption units for different types of meters. When the units are monetary, then the name is expressed using the 3 character international symbol (GBP, USD, etc.) as defined in ISO 4217.

```
currency::= structure
{
    currency_name:      utf8-string,
    currency_scale:     integer,
    currency_unit:      enum
}
```

Where:

currency_name: utf8-string

as per ISO 4217

OR

min = minutes,

hrs = hours,

sec = seconds,

kWh = kilowatt hours,

Whr = Watt hours,

mt3 = m^3,

ft3 = ft^3,

Jle = Joule

OR

Defined by project specific companion specification such that all characters used are lower case.

The *currency_scale* indicates the exponent of a decimal multiple or submultiple of the base unit in which the relevant attribute values are expressed. Negative values indicate submultiples.

NOTE For example: if the currency is Euros and the values are expressed in tenths of one cent (one thousandth of a Euro) then the scaler will have value -3 (minus 3).

currency_unit: enum:
(0) time,
(1) consumption,
(2) monetary

The *currency_unit* is an enumerated value that indicates the generic type of currency:

– time (in minutes, seconds, hours etc.);

– consumption (in kWhrs, Whrs, m^3, Whrs etc.); or

monetary (GBP, USD, EUR, etc.).

4.6.2.2.16 low_credit_threshold

This attribute reflects the *warning_threshold* attribute of the “Credit” object currently *In use*. This threshold can be used to generate a warning to the consumer, for example. It has no other function.

double-long, scaled according to the *currency* attribute.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE The value of this attribute will change depending on which "Credit" object is *In use* at the time of the query.

4.6.2.2.17 next_credit_available_threshold

This threshold reflects the credit_available_threshold attribute of the next-priority "Credit" object (except when there is no next priority "Credit"; in which case this attribute has the value -2 147 483 648 (largest possible negative value)).

When the *available_cre*dit attribute of the "Account" object becomes less than or equal to this value, the *credit_status* attribute of the next priority "Credit" object changes to:

(1) *Selectable* in the case when the *credit_configuration* attribute of the "Credit" object concerned has the bit 1 (Requires confirmation) set;

(2) *Selected/Invoked* in the case where the bit 1 (Requires confirmation) is cleared and the next_credit_available_threshold is greater than zero;

(3) *In use* in the case where the bit 1 (Requires confirmation) is cleared and the next_credit_available_threshold is equal to zero.

NOTE The "next-priority" refers to the next "Credit", in priority order, that has not yet been selected. This attribute will change depending on which "Credit" is next in priority order.

The *next_credit_available_threshold* will change depending on which "Credit" is *In use*.

double-long, scaled according to the *currency* attribute.

4.6.2.2.18 max_provision

This attribute affects the operation of "Charge" objects that are configured with *charge_type* equal to (2) *payment_event_based_collection*.

This attribute holds the limit amount scaled as defined in the *currency* attribute across all "Charge" objects with *charge_type* (2) *payment_event_based_collection.* The limit is related to the time period defined in *max_provision_period.*

NOTE 1 The combination of this attribute and *max_provision_period* are intended to provide a limit to the total value of collections from top-ups within, for example, one week.

NOTE 2 If a consumer vends many times in a short period, it is possible that he pays more than is required by the utility into one of his instantiated "Charge" objects (due to the unpredictability of payment event based collection).

long-unsigned, scaled according to the *currency* attribute.

4.6.2.2.19 max_provision_period

This attribute holds the time period applicable for the *max_provision* attribute. This is specified in seconds.

double-long

4.6.2.3 Method description

4.6.2.3.1 activate_account (data)

This method allows the activation of the "Account". The *account_status* element of *account_mode_and_status* will be set to (2) *Account active*.

NOTE The *account_activation_time* will be set to the time of invoking this method.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.3.2 close_account (data)

This method forces the closure of the Account.

NOTE 1 This may be done pursuant to a change of supplier or change of tenancy or any other reason.

The *account_status* element of *account_mode_and_status* will be set to (3) *Account closed*. The *account_closure_time* is set to the time of invoking this method.

NOTE 2 When this method is invoked the Account will no longer be active. As such its attributes will cease to change along with the attributes of all "Credit" and "Charge" objects listed within the *credit_reference_list* and *charge_reference_list* until such time that the "Account" object becomes active again, or the "Credit" and "Charge" objects are referenced from another "Account" object.

data::= integer (0)

4.6.2.3.3 reset_account (data)

If the *account_status* of the attribute *account_mode_and_status* is not equal to (2) *Active account*, then this method sets the attributes of the "Account" to default values except where an attribute does not have a default value, in which case the behaviour shall be project specific.

If the *account_status* element of the attribute *account_mode_and_status* is equal to 1 (New, inactive account), then this method shall have no effect.

data::= integer (0)

4.6.3 Credit interface class (class_id = 112, version = 0)

4.6.3.1 General

Instances of the "Credit" IC allow the management of a credit that can be consumed by charges. There are several different credit types; each "Credit" object characterizes itself by the values of its attributes.

All "Credits" associated with one supply are listed in the *credit_reference_list* attribute of the "Account" object. "Credits" move between states by:

- top-ups;
- the adjustment of current_credit_amount by method invocation; or
- the decrement of credit by charges.

This is explained in the state diagram in 4.6.3.2. Subclause 4.6.1 lists "Credit" types as defined in IEC TR 62055-21:2005.

4.6.3.2 Credit states

The credit states only have meaning when *priority* is non-zero. They are shown in Table 34 and Figure 24. The state transitions are shown in Table 35.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 34 – Credit states

Priority	Credit state	Meaning
0	Any	The instance of the "Credit" object is inactive. NOTE When a "Credit" has a non-zero priority, but does not appear in any *credit_reference_list* then it has the same behaviour as if it had a zero priority.
>0	(0) Enabled	Reference to the "Credit" appears in the *credit_reference_list* of an active "Account" with a non-zero priority.
>0	(1) Selectable	The "Credit" requires some additional interaction before it can be *In use*. A credit is selectable only when it is the next priority credit, it is not exhausted and when bit 1 (Requires confirmation) of "Credit" *credit_configuration* is set.
>0	(2) Selected/Invoked	The "Credit" that was selectable has now been selected but it may not yet be *In use* (it could be that some of the higher priority "Credit" is still being used i.e. in the case of EMC). Alternatively a "Credit" that was Enabled and did not require selection may arrive here directly if the higher priority credit has become *Exhausted*.
>0	(3) In use	The "Credit" is being used to pay *Charge*s within the meter.
>0	(4) Exhausted	The "Credit" has run out.

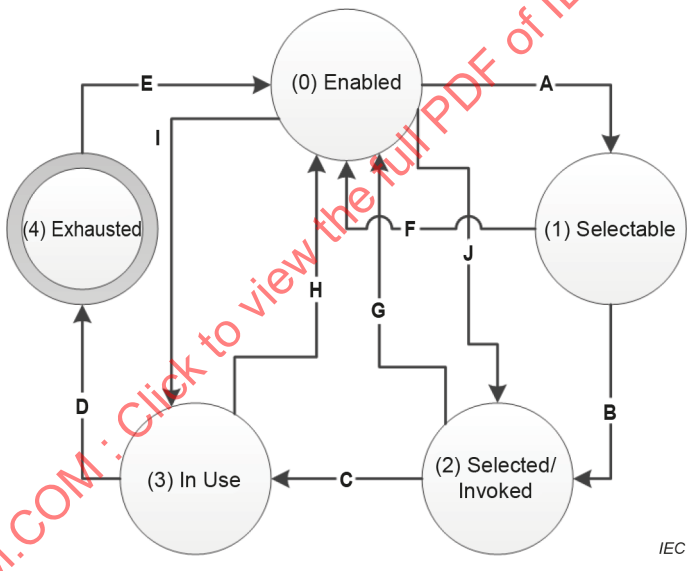


Figure 24 – Credit States when priority >0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 35 – Credit state transitions

Credit state	From	To	Conditions
A	(0) Enabled	(1) Selectable	A “Credit” object becomes selectable when it is the highest priority “Credit” object that is not exhausted and when the “Account” object *available_credit* reaches the “Account” object *next_credit_available_threshold*. NOTE 1 This only applies when the *credit_configuration* attribute of the “Credit” object concerned has bit 1 (Requires confirmation) set.
B	(1) Selectable	(2) Selected/ Invoked	The trigger to the transition is external to the COSEM domain such as a button push on the meter or some other stimulus. NOTE 2 This only applies when the *credit_configuration* attribute of the “Credit” object concerned has bit 1 (Requires confirmation) set.
C	(2) Selected/ Invoked	(3) In use	This transition occurs when the previous priority “Credit” object becomes exhausted. NOTE 3 At this point the “Account” object *next_credit_available_threshold* updates to reflect the *credit_available_threshold* of the next priority “Credit” object.
D	(3) In use	(4) Exhausted	This transition occurs when the *current_credit_amount attribute of the “Credit” object* reaches the level set in the *limit* attribute. NOTE 4 This may indirectly cause a disconnection of supply, in the case where there is no lower priority “”Credit” object to become *In use*. NOTE 5 At the point when the current “Credit” becomes exhausted, *credit_status* is set to(4) *Exhausted*.
E	(4) Exhausted	(0) Enabled	This transition occurs when the *current_credit_amount* becomes larger than the *limit* attribute., or – in the case of a repayable credits – the credit used has been paid back. NOTE 6 This “Credit” object has received some top-up amount from an incoming token or method invocation.
F	(1) Selectable	(0) Enabled	This transition occurs when the “Account” object *available_credit* becomes larger than the “Account” object *next_credit_available_threshold*. NOTE 7 This only applies when the *credit_configuration* attribute of the “Credit” object concerned has the bit 1 (Requires confirmation) set. NOTE 8 This is usually because the *current_credit_amount* attribute of a “Credit” object that is *In use* has been increased.
G	(2) Selected / Invoked	(0) Enabled	This transition occurs when the “Account” object *available_credit* becomes larger than the “Account” object *next_credit_available_threshold*. NOTE 9 This is usually because the *current_credit_amount* attribute of a “Credit” object that is *In use* has been increased.
H	(3) In use	(0) Enabled	This transition occurs when *the credit_status* of a higher priority “Credit” object becomes *In use*. NOTE 10 This is usually because the higher priority “Credit” object *current_credit_amount* has been increased. At this point the *next_credit_available_threshold* of the “Account” object will also change.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Credit state	From	To	Conditions
I	(0) Enabled	(3) In use	This transition occurs when the immediately higher priority "Credit" object becomes exhausted. NOTE 11 This only applies when the *credit_configuration* attribute of the "Credit" object concerned has the bit 1 (Requires confirmation) cleared and the *credit_available_threshold* of the "Credit" object is set to zero and it's *current_credit_amount* is greater than the *limit* (a credit cannot be *In use* until the higher priority Credit is Exhausted).
J	(1) Enabled	(2) Selected/ Invoked	The transition occurs when the *available_credit* in the "Account" object becomes less than the *next_credit_available_threshold* on the "Account" object. NOTE 12 This only applies when the *credit_configuration* attribute of the "Credit" object concerned has the bit 1 (Requires confirmation before it can be selected or *In use*) cleared, and the *credit_available_threshold* of this "Credit" object is greater than zero.

4.6.3.3 Current credit status flags

The significance of the *current_credit_status* flags (this is held by the *current_credit_status* attribute of the "Account" object) is explained in Figure 25.

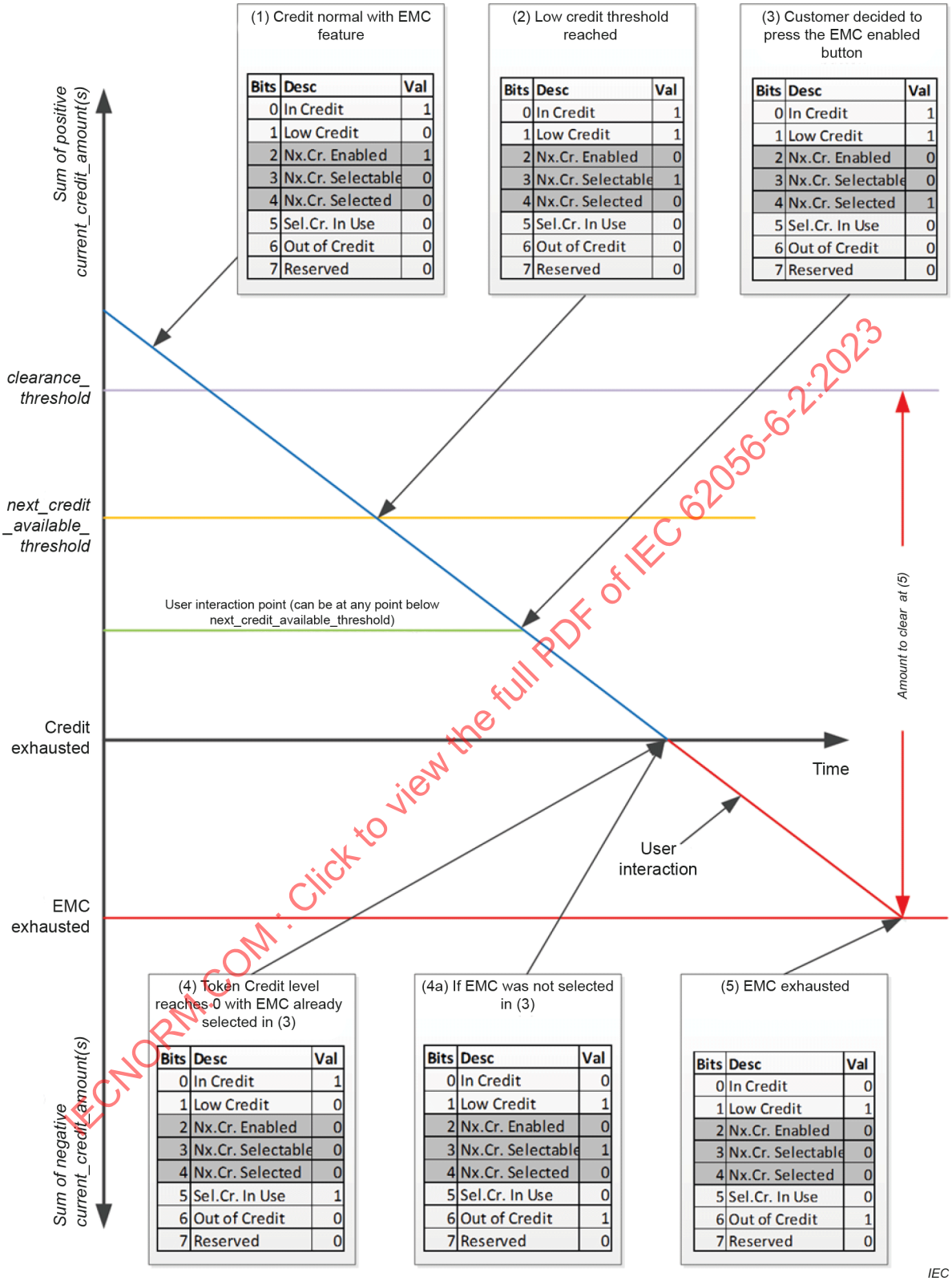
In Figure 25 an example is given for possible cases involving a payment metering system with two "Credit" objects: Token Credit and Emergency Credit.

NOTE This is one possible implementation. The type and number of credits used in an actual project are subject to project specific companion specifications.

There are two options for selection of selectable credits; either selection of a "Credit" whilst the current "Credit" is *In use* (case 4) or when the current "Credit" has been exhausted (case 4a).

In the case of credit being selected while current credit is *In use* (case 4), credit will start to be consumed from the next credit immediately after current credit has exhausted. In the case (4a) the current credit will become exhausted and potentially continue to be decremented by charges until the consumer takes action by selecting the selectable credit or adding a top up.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



Bits	Desc	Val
0	In Credit	1
1	Low Credit	0
2	Nx.Cr. Enabled	1
3	Nx.Cr. Selectable	0
4	Nx.Cr. Selected	0
5	Sel.Cr. In Use	0
6	Out of Credit	0
7	Reserved	0

Bits	Desc	Val
0	In Credit	1
1	Low Credit	1
2	Nx.Cr. Enabled	0
3	Nx.Cr. Selectable	1
4	Nx.Cr. Selected	0
5	Sel.Cr. In Use	0
6	Out of Credit	0
7	Reserved	0

Bits	Desc	Val
0	In Credit	1
1	Low Credit	1
2	Nx.Cr. Enabled	0
3	Nx.Cr. Selectable	0
4	Nx.Cr. Selected	1
5	Sel.Cr. In Use	0
6	Out of Credit	0
7	Reserved	0

Bits	Desc	Val
0	In Credit	1
1	Low Credit	0
2	Nx.Cr. Enabled	0
3	Nx.Cr. Selectable	0
4	Nx.Cr. Selected	0
5	Sel.Cr. In Use	1
6	Out of Credit	0
7	Reserved	0

Bits	Desc	Val
0	In Credit	0
1	Low Credit	1
2	Nx.Cr. Enabled	0
3	Nx.Cr. Selectable	1
4	Nx.Cr. Selected	0
5	Sel.Cr. In Use	0
6	Out of Credit	1
7	Reserved	0

Bits	Desc	Val
0	In Credit	0
1	Low Credit	1
2	Nx.Cr. Enabled	0
3	Nx.Cr. Selectable	0
4	Nx.Cr. Selected	0
5	Sel.Cr. In Use	0
6	Out of Credit	1
7	Reserved	0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 25 – Operation of current_credit_status flags

Credit		0...n	class_id = 112, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. current_credit_amount	(dyn.)	double-long				x + 0x08
3. credit_type	(static)	enum				x + 0x10
4. priority	(static)	unsigned				x + 0x18
5. warning_threshold	(static)	double-long				x + 0x20
6. limit	(static)	double-long				x + 0x28
7. credit_configuration	(static)	bit-string				x + 0x30
8. credit_status	(dyn.)	enum				x + 0x38
The following attribute (9) applies to emergency, some time-based (could be reserved credit), and consumption based credits only.						
9. preset_credit_amount	(static)	double-long				x + 0x40
10. credit_available_threshold	(static)	double-long				x + 0x48
The following attribute applies to time-based, and consumption based credit only.						
11. period	(static)	date-time				x + 0x50
Specific methods		***m/o***				
1. update_amount (data)		o				x + 0x58
2. set_amount_to_value (data)		o				x + 0x60
The following method applies to emergency, and time-based consumption based credit only.						
3. invoke_credit (data)		o				x + 0x68

4.6.3.4 Attribute description

4.6.3.4.1 logical_name

Identifies the “Credit” object instance. See 6.2.17.

4.6.3.4.2 current_credit_amount

Provides the credit value of this particular “Credit” object. (See Figure 26).

NOTE This value is increased and decreased by invoking the methods of this object, the action of top-ups, and the collection of charges. This value contributes to the *available_credit* attribute in the “Account” object from which the “Credit” object is referenced.

double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.4.3 credit_type

This is an enumeration which identifies the type of Credit that this object represents. The type indicates which attributes are expected to be processed for this “Credit” object, but the actual processing is directed by values of other attributes including the configuration flags. The types available are:

enum:

(0) token_credit,
(1) reserved_credit,
(2) emergency_credit,
(3) time_based_credit,
(4) consumption_based_credit

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.3.4.4 priority

Describes the activation priority of this “Credit” object.

Value 1 is the highest priority and value 255 the lowest.

Every “Credit” object shall have a different priority, except in the case of priority 0.

NOTE Behaviour will be undefined if there are multiple “Credit” objects configured with the same non-zero priority.

A value of 0 indicates that the “Credit” object is never activated, although it can still be configured and its *current_credit_amount* can be adjusted by invoking its methods, but it cannot receive top ups.

When a “Credit” object of priority zero is configured it shall not appear in the “Account” *credit_reference_list*, it will be not considered as part of the *available_credit* calculation and it shall not be decremented by charges.

See the *credit_reference_list* attribute of the “Account” object (4.6.2.2.9) for more information.

4.6.3.4.5 warning_threshold

Holds a threshold value for *current_credit_amount.* When *current_credit_amount* is decremented to the value *of warning_threshold*, a warning is triggered for the consumer that credit is low.

NOTE The *low_credit_theshold* attribute of the associated “Account” object echoes this attribute of the currently *In use* “Credit” object.

double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.4.6 limit

This attribute holds a threshold value for *current_credit_amount*. When *current_credit_amount* is decremented to the value of *limit*, then *credit_status* becomes (4) *Exhausted*.

NOTE This is typically set to zero in a meter operating in prepayment mode. For a meter operating in credit mode the highest-priority “Credit” object would normally have a limit equal to the largest possible negative number.

double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.4.7 credit_configuration

Allows configuring the behaviour of the “Credit” object.

If **bit 0** is set then the credit item requires a visual indication on the meter display when it becomes selected.

If **bit 1** is set, confirmation is required from the consumer before moving into this type of “Credit” (for example this is the case of an emergency credit object, where the consumer is required to press a button before the “Credit” is (2) Selected/Invoked).

If **bit 2** is set then this indicates that this “Credit” amount requires repayment and consequently the credit used will contribute to the *amount_to_clear* attribute in the “Account” object that references the particular “Credit” object.

If **bit 3** is set then the *current_credit_amount* can be cleared by an end of billing period action (using a script acting upon the *set_amount_to_value* method. It is not recommended to configure “Credits” to permit this unless the meter is operating in credit mode.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If **bit 4** is set then this “Credit” object will be permitted to receive credit from tokens.

credit_configuration: bit-string:

Bit 0 = Requires visual indication,

Bit 1 = Requires confirmation before it can be selected/invoked, or

Bit 2 = Requires the credit amount to be paid back,

Bit 3 = Resettable,

Bit 4 = Able to receive credit amounts from tokens.

4.6.3.4.8 credit_status

Driven by the prepayment application to indicate the state that the “Credit” object is in.

The states appear in Figure 24 above. Depending on the type of the “Credit” and its configuration, the value of this attribute is driven by the application based on the values of the *current_credit_amount*, *limit*, *preset_credit_amount* and *credit_available_threshold* attributes of the “Credit” objects and the *amount_to_clear* attribute of the “Account”.

When the *amount_to_clear* attribute of the ‘Account” object referencing this “Credit” object transitions from a negative *value* to zero the EMC status shall be (0) *Enabled* again.

NOTE When a “Credit” object has a *credit_status* (4) *Exhausted* then this “Credit” cannot be invoked again until the “Credit” transitions to (0) *Enabled*. If this is the lowest priority “Credit” object, then charges may still be applied to this object.

enum:

(0) The instance of the credit class is in the state of *Enabled*,

(1) The instance of the credit class is in the *Selectable* state,

(2) The instance of the credit class is in the *Selected/Invoked* state,

(3) The instance of the credit class is in the *In use* state and may be consumed by charges,

(4) The *current_credit_amount* has been entirely consumed and the credit is considered *Exhausted*.

For additional explanation see Table 34.

4.6.3.4.9 preset_credit_amount

This attribute is a value that is set as an initial amount of credit that is available for the “Credit” object.

The value is added to the *current_credit_amount* attribute:

a) when the *credit_status* becomes (2), *Selected/Invoked* if *credit_configuration* bit 1 (Requires confirmation) is set, and the application confirmation occurs, or

b) when the *credit_status* becomes (3) *In use* if *credit_configuration* bit 1 (Requires confirmation) is cleared unless the *credit_status* is (4) *Exhausted*, or

c) when the method *invoke_credit* is invoked, if the *credit_configuration* bit 2 (Requires the credit amount to be paid back) is set, or

d) when the date_time in *period* occurs which is implicit.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

When a “Credit” does not require a preset amount, the *preset_credit_amount* shall be 0 and the “Credit” can receive credit amounts from credit tokens or by invocation of the *update_amount* and *set_amount_to_value* methods. The value of *preset credit_amount* does not change unless a new value is written by the client.

In the case of “Credits” of type *emergency_credit*:

– the *preset_credit_amount* attribute shall be used,
– the “Credit” shall only be able to receive credit amounts from tokens when:
  - the status is (3) *In use or* (4) *Exhausted*;
  - some (or all) credit has been used; and
  - it is required to be paid back (*credit_configuration* has bit 2 (Requires the credit amount to be paid back) set.

NOTE When the *current_credit_amount* has reached the *limit* then the *credit_status* attribute will become (4) *Exhausted*. If the *credit_configuration* bit 2 (Requires the credit amount to be paid back) is set, then the credit used is the difference between *preset_credit_amount* and *current_credit_amount* (positive value), and it would be usual that it is repaid by the addition of a credit token or by means of invoking a method. After paying back the “Credit” the *current_credit_amount* will be zero but the *credit_status* will be (0) Enabled in the case of *amount_to_clear* = 0 or (4) Exhausted in the case where *amount_to_clear* is less than zero.

If this attribute is set to zero there is no behaviour associated with it.

double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.4.10 credit_available_threshold

A threshold value related to the “Account” object *available_credit.*

When the *available_credit* on the “Account” object decrements to the *credit_available_threshold* on the next-priority “Credit” object, the *credit_status* of that object changes to:

a) 1 (Selectable) if the *credit_configuration* bit 1 (Requires confirmation) is set, or
b) 2 (Selected/Invoked) if the *credit_configuration* bit 1 (Requires confirmation) is cleared.

NOTE 1 This reflects whether the “Credit” requires confirmation before it is put into use.

NOTE 2 A “Credit” will not be *In use* until exhaustion of a higher-priority “Credit”, but if selected before a higher priority “Credit” is exhausted the value of the *current_credit_amount* attribute will contribute to *available_credit* in the associated “Account”.

double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.4.11 period

Where the *credit_type* = 3 (time_based_credit) or *credit_type* = 4 (consumption_based_credit), this attribute holds the time at which the *current_credit_amount* is set automatically to the *preset_credit_amount*.

octet-string, formatted as specified in 4.1.6.1 for *date_time.* Wildcards are allowed. If all fields are wildcards, the *preset_credit_amount* will never be added.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.3.5 Method description

4.6.3.5.1 update_amount (data)

This method adjusts the value of the *current_credit_amount* attribute. Positive values adjust the *current_credit_amount* positively. Negative values are also permitted.

data::= double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.5.2 set_amount_to_value (data)

This method sets the value of the *current_credit_amount* attribute. The amount previously in the updated attribute shall be given as the response parameter.

data::= double-long, scaled according to the *currency* attribute of the “Account” object.

Returns double-long, scaled according to the *currency* attribute of the “Account” object.

4.6.3.5.3 invoke_credit (data)

This method is invoked to bring this “Credit” object to *credit_status* (2)*Selected/Invoked* if it has a *credit_configuration* bit 1 is set (Requires confirmation), and the *credit_status* is (1) *Selectable*.

The mechanism for selecting the “Credit” is not in the scope of COSEM and shall be specified by the implementer (e.g. button push, meter process, script, etc.)

data::= integer (0)

4.6.3.6 Additional remarks

a) Although it is usual that tokens cause the incrementing of *current_credit_amount* and application of “Charge” objects cause decrement, these inputs are applied by means of signed addition and it could be possible to accept tokens or “Charge” objects with negative values.

b) Behaviour of the emergency_credit type is determined by the *preset_credit_amount* attribute and the ‘Requires the credit amount to be paid back’ option in *credit_configuration* attribute.

When the “Credit” object is selected/invoked, the value of the *preset_credit_amount* attribute is added to the value of the *current_credit_amount* attribute (which is normally zero at this point in its lifecycle).

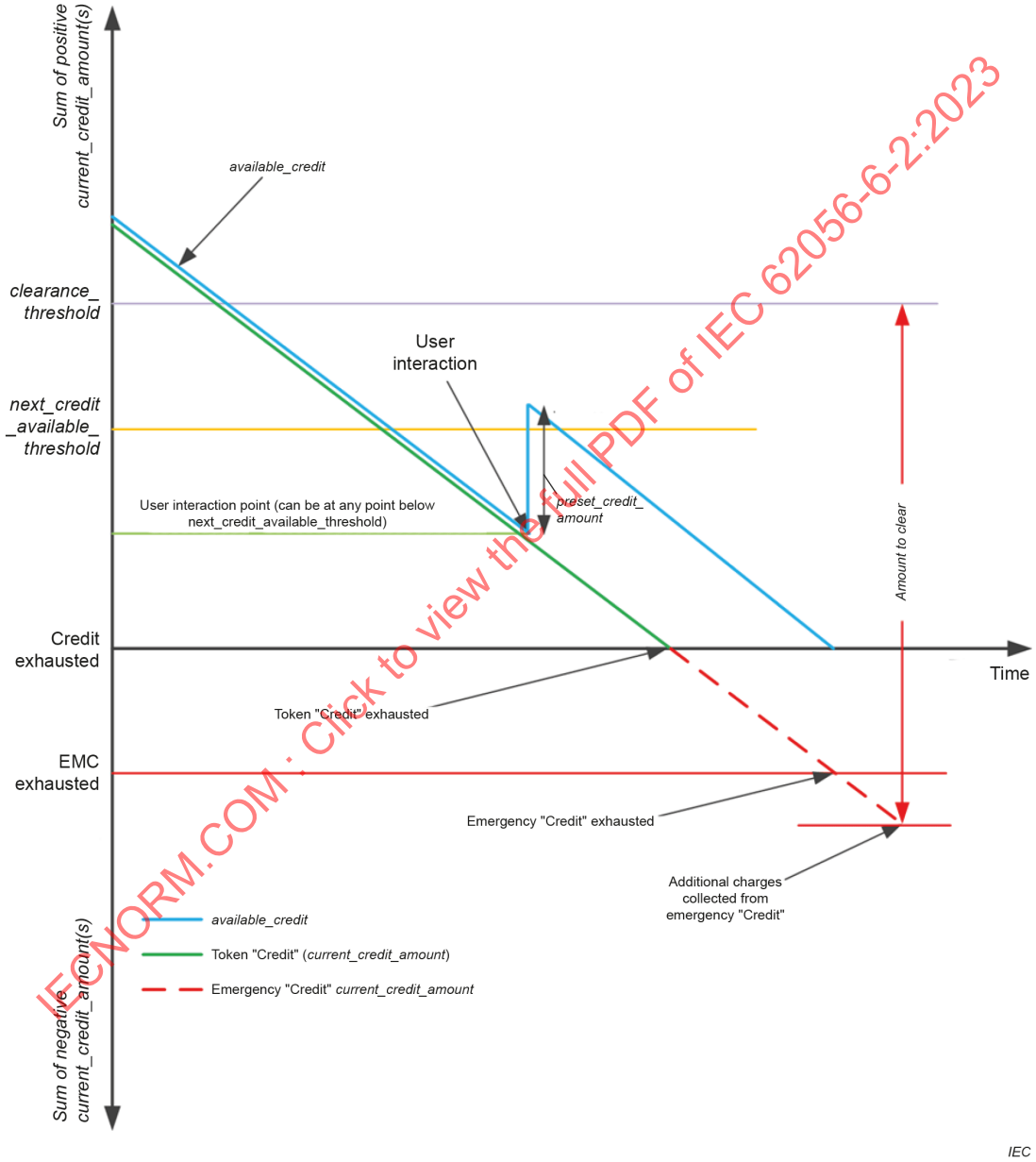
c) To replenish emergency credit and make its status (0) Enabled after being (3) *In use*, sufficient payment should be received to take the *available_credit* of the “Account” object up to at least the *clearance_threshold.*

This will necessarily involve providing sufficient funds to repay the negative value of *current_credit_amount* of the “Credit” objects providing EMC function. To allow for the joint management of multiple “Credit” instances the credit used values for each “Credit” object are aggregated into the *amount_to_clear* together with the *clearance_threshold* attribute of the relevant “Account” object. When *amount_to_clear* reaches zero the status (4) *Exhausted* is by definition cleared on all associated “Credit” objects.

d) The “Account” object *token_gateway_configuration* attribute determines the distribution of credit to “Credit” objects.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 26 shows interaction of the *current_credit_amount* attributes of two “Credit” Objects in a payment metering system. It can be seen that the Emergency Credit, when selected contributes to the *available_credit* but does not become *In use* until the Token “Credit” has become *Exhausted*. At the point where Emergency “Credit” is *In use* it starts contributing to the *amount_to_clear*. When the Emergency “Credit” is *In use* and contributing to *amount_to_clear*, the *amount_to_clear* also takes into consideration the clearance threshold. The clearance threshold is only important when Token “Credit” *curren_credit_amount* is zero or below.



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 26 – Interaction of current_credit_amount and available_credit with Token “Credit” and Emergency “Credit”

As a result of invoking the Emergency “Credit” the *available_credit* will become bigger than *next_credit_available_threshold* but the Emergency “Credit” will not change *credit_status* from (1) *Selected* to (0) *Enabled.* However if the top up or method invocation to *set_amount_to_value*, forced the *available_credit* (by means of increasing the *current_credit_amount* of a “Credit” object) to be more than *next_credit_available_threshold* then the status shall be forced to (0) *Enabled.*

4.6.4 Charge (class_id = 113, version = 0)

4.6.4.1 General

Instances of the “Charge” IC allow the management of a single Charge. Depending on the attributes configured such as amount per price and the period, the Charge is taken at appropriate times from the “Credit” object *In use*.

NOTE The details of the collection (charge-taking) cycle may be project dependent, and thus they are subject to project specific companion specifications since they affect third-party estimates of current values.

Each “Charge” object characterises itself by the values of its attributes.

All “Charges” associated with one supply are referenced in the *charge_reference_list* attribute of the related “Account” object (4.6.2.2.10).

Charge		0...n	class_id = 113, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. total_amount_paid	(dyn.)	double-long				x + 0x08
3. charge_type	(static)	enum				x + 0x10
4. priority	(static)	unsigned				x + 0x18
5. unit_charge_active	(static)	structure				x + 0x20
6. unit_charge_passive	(static)	structure				x + 0x28
7. unit_charge_activation_time	(static)	octet-string				x + 0x30
The following attribute relates to time based and consumption based collection only.						
8. period	(static)	double-long-unsigned				x + 0x38
The following attributes relate to all charge types.						
9. charge_configuration	(static)	bit-string				x + 0x40
10. last_collection_time	(dyn.)	date-time				x + 0x48
11. last_collection_amount	(dyn.)	double-long				x + 0x50
12. total_amount_remaining	(dyn.)	double-long				x + 0x58
The following attributes relate to payment event based collection only.						
13. proportion	(static)	long-unsigned				x + 0x60
Specific methods		***m/o***				
1. update_unit_charge (data)		o				x + 0x68
2. activate_passive_unit_charge (data)		o				x + 0x70
The following methods relate to time-based and payment event based collection only .						
3. collect (data)		o				x + 0x78
The following methods relate to payment event based collection only.						
4. update_total_amount_remaining (data)		o				x + 0x80
5. set_total_amount_remaining (data)		o				x + 0x88

4.6.4.2 Attribute description

4.6.4.2.1 logical_name

Identifies the “Charge” object instance. See 6.2.17.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.4.2.2 total_amount_paid

Holds the total amount collected for this ”Charge” object. It is not normally reset while the “Account” remains active.

NOTE This value is incremented by the application at the same time and at the same rate as the charge that is collected.

4.6.4.2.3 charge_type

Designates the type of collection associated with this instance of the “Charge” class.

enum:

(0) consumption_based_collection,
(1) time_based_collection,
(2) payment_event_based_collection

The *charge_type* indicates which attributes are expected to be processed for this “Charge” object. The processing is also influenced by the *charge_configuration* attribute.

NOTE In the case of *payment_event_based_collection* charges this attribute dictates the mechanism of charge collection. The application collects *payment_event_based_collection* charges from appropriate “Credits” as soon as credit is distributed from a new token.

4.6.4.2.4 priority

Describes the priority of this particular “Charge” when making collection from a “Credit” object.

Every “Charge” object shall have a different priority, except in the case of priority 0.

A value of 1 represents highest priority and a value of 255 the lowest.

A value of 0 indicates that the “Charge” is not activated, though it can still be configured and its *total_amount_remaining* can be adjusted by invoking the appropriate methods of the “Charge” object.

“Charge” objects with priority value 0 shall not appear in the *charge_reference_list* of the “Account” object, but “Charge” objects with *priority* <> 0 do necessarily appear in the *charge_reference_list* of an “Account”.

4.6.4.2.5 unit_charge_active

Defines the active price, that is the amount charged per unit consumed, per unit time, or per payment received, in terms of the currency attribute of the relevant “Account” instance and where relevant, the *scaler_unit* attribute of the object identified by the *commodity_reference* structure.

```
unit_charge_active::= structure
{
    charge_per_unit_scaling:    charge_per_unit_scaling_type,
    commodity_reference:        commodity_reference_type,
    charge_table:               charge_table_type
}
```

Where:

```
charge_per_unit_scaling_type::= structure
{
    commodity_scale :    integer,
    price_scale:         integer
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
commodity_reference_type::=      structure
{
        class_id:                long-unsigned,
        logical_name:            octet-string,
        attribute_index:         integer
}

charge_table_type::= array   charge_table_element

charge_table_element::=      structure
{
        index:                   octet-string,
        charge_per_unit:         long
}
```

a) Explanations related to charge_per_unit_scaling

Where the *charge_type* = (0) *consumption_based_collection* the field in *charge_per_unit_scaling* is expressed as the exponent (to the base of 10) of the multiplication factor of the base unit in which the relevant attribute values are expressed internally:

– commodity_scale is applied to the units associated with *commodity_reference*,

– price_scale is applied to the *currency_scale* of the *currency* attribute of the relevant "Account" object.

Where the *charge_type* IS NOT (0) *consumption_based_collection* , the *commodity_scale* shall be set to 0 and the *price_scale* is expressed as the exponent (to the base of 10) of the *currency_scale* to be applied to the *currency* attribute of the "Account" object.

NOTE 1 For example, suppose the primary *currency_name* is Euros (as determined in the "Account" object) with values expressed in units of one cent (one hundredth of a Euro) and the commodity being supplied is active import electrical energy with values expressed in units of 1 000 Wh (one kWh) from the *scaler_unit* attribute of the *commodity_reference*). Then for a price in tenths of a cent per kilowatt-hour the price scale will have value –3 (minus-three) since it is in thousandths of a base unit and the commodity scale will have value 0 (zero) since it is in base units (i.e. kWh).

For the case where the primary currency is kWh or similarly related to the *commodity_scale* then a TOU tariff is not possible and as such the *charge_table* element of the *unit_charge_active* and *unit_charge_passive* shall not be relevant but the *commodity_scale* and *price_scale* shall be the same value.

NOTE 2 The *commodity_scale* and *price_scale* do not imply any particular rate of collection.

b) Explanations related to *charge_type*

When the *charge_type* = (0) *consumption_based_collection* the *commodity_reference* identifies a *scaler_unit* attribute of a total register, measuring whatever is being supplied, for instance, electrical import energy.

When the charge_type = (1) time_based_collection or (2) payment_event_based_collection, then the commodity_reference shall be a structure of zero elements.

c) Explanations to charge_table elements

Each element in the array *charge_table* is a structure identifying what is being charged for and collected.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where charge_type = (0) consumption_based_collection:

– index is an identifier of a derivative of the main commodity reference, for example tariff rate registers,
– charge_per_unit is the charge amount per unit that is scaled by the *charge_per_unit_scaling* factors.

NOTE 3 For example, if the end consumer generates energy then the “Charge” should be negative, meaning that the utility pays the end consumer for the energy he produces.

NOTE 4 It is also possible to model exported energy using a separate “Account” object and other related objects.

Where the *charge_type* = (1) *time_based_collection* or (2) *payment_event_based_collection*, then a single entry into the array *charge_table* is made, containing:

– index is an octet-string of length 0;
– charge_per_unit is a value equivalent to the amount charged per *period* or per payment event that is scaled by the *charge_per_unit_scaling* factors.

4.6.4.2.6 unit_charge_passive

Holds the details of prices to be applied at the occurrence of the activation date.

Its data structure is the same as the *unit_charge_active*. On activation, the whole structure of *unit_charge_passive* is copied into *unit_charge_active*.

4.6.4.2.7 unit_charge_activation_time

Defines the time when the object itself invokes the specific method *activate_unit_charge_passive*.

A definition with "not specified" notation in all fields of the attribute will deactivate this automatism. Partial "not specified" notation in just some fields of date and time is not allowed.

octet-string, formatted as specified in 4.1.6.1 for *date_time*.

4.6.4.2.8 period

Where *charge_type* = (0) *consumption_based_collection* or (1) *time_based_collection* it defines the period over which the Charge will be collected. If period is set to zero there is no automatic collection of this “Charge”: collection is done by invoking the *collect* method.

NOTE Implementations may vary: it is possible to collect the whole charge in a single collection, or to divide the collection into parts.

Where charge_type = (2) payment_event_based_collection this attribute is not used.

double-long-unsigned, expressed in seconds.

4.6.4.2.9 charge_configuration

This attribute is a bit-string defined as follows:

charge_configuration::= bit-string

Bit 0 = Percentage based collection,

Bit 1 = Continuous collection.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If bit 0 is set and *charge_type* = (2) *payment_event_based_collection* then this "Charge" object will collect a proportion of each top-up in accordance with the proportion attribute (4.6.4.2.13). This applies only to token top ups and not to method invocations.

NOTE The proportion attribute of the "Charge" object and the provision attribute of the "Account" object provide more information on this collection.

If bit 1 is set then collection will not terminate when the *total_amount_remaining* is equal to zero. If bit 1 is cleared then charge collection will terminate when the *total_amount_remaining* attribute is equal to zero.

4.6.4.2.10 last_collection_time

Holds the *date_time* when the last collection took place.

This includes incremental collection made against consumption.

octet-string, formatted as specified in 4.1.6.1 for *date_time*.

4.6.4.2.11 last_collection_amount

Holds the last collection amount relating to the *last_collection_time* attribute.

double-long, scaled according to the *price_scale* element of *unit_charge_active*.

4.6.4.2.12 total_amount_remaining

Where *charge_configuration* bit 1 (Continuous collection) is cleared, this attribute holds the total amount remaining for this "Charge" object.

NOTE 1 The value of this attribute is initially configured by invoking the *set_total_amount_remaining* method.

NOTE 2 This attribute is used to limit the collections for repayment of a debt. It is not a direct measure of the consumer's liability. The value is not allowed to go negative (it is set to zero instead) and is not incremented by collection of a "Charge" with a negative price.

Where *charge_configuration* bit 1 (Continuous collection) is set, this attribute is zero.

NOTE 3 See also the Additional Notes at the end of the "Charge" IC specification.

double-long, scaled according to the *price_scale* element of *unit_charge_active*.

4.6.4.2.13 proportion

Where the *charge_configuration* bit 0 (Percentage based collection) is set and *charge_type* = (2) *payment_event_based_collection*, then this attribute is the proportion of each top-up amount processed within the "Token gateway" object linked via the "Account" object to this "Charge" object.

The range 0x0000 to 0x2710 (0 to 10,000) represents 0 to 100 %.

NOTE The amount of collection that occurs may be limited by the *max_provision* and *max_provision_period* attributes of the "Account" object.

If a fixed amount is required to be collected at every top-up (*charge_type* = 2), then the amount to be taken should be set in the first element of the *charge_table* array in the *unit_charge_active* / *unit_charge_passive* attribute.

Where either *charge_type* <> (2) *payment_event_based_collection* or *charge_configuration* bit 0 (Percentage based collection) is cleared then this attribute has no effect.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

In the case when *charge_type* = (2) *payment_event_based_collection* and the collection of a fixed amount is required see *unit_charge_active* (4.6.4.2.5) and *unit_charge_passive* (4.6.4.2.6).

4.6.4.3 Method description

4.6.4.3.1 update_unit_charge (data)

Allows updating a subset of unit charge values inside the *unit_charge_passive* attribute. The *charge_per_unit_scaling* and commodity_reference values are not affected by this method.

It remains necessary to activate the *unit_charge_passive* so that the values can take effect.

```
data::= array charge_table_element
charge_table_element::=  structure
{
        index:                  octet-string,
        charge_per_unit:        long
}
```

See the *charge_table_element* of the *unit_charge_active* attribute for a description of the elements.

NOTE In the case where the index exists then the value is overwritten, and if the index does not exist then the new value is appended to the array.

4.6.4.3.2 activate_passive_unit_charge (data)

Copies the whole structure of the *unit_charge_passive* attribute into the *unit_charge_active* attribute.

NOTE This method has no effect on the collection activity or the priority of the “Charge” object.

data::= integer (0)

4.6.4.3.3 collect (data)

Where *charge_type* <> (0) *consumption_based_collection*, this method makes collection of the amount defined in *unit_charge_active* when *charge_configuration* bit 0 (percentage based collection) is cleared.

Where *charge_type* = (0) *consumption_based_collection*, this method has no effect.

data::= integer (0).

4.6.4.3.4 update_total_amount_remaining (data)

Allows the update of the *total_amount_remaining* attribute. The value is to be scaled according to the *price_scale* of *unit_charge_active*. The value is added to *total_amount_remaining*. The amount previously in *total_amount_remaining* attribute shall be given as the response parameter.

NOTE This does not affect *total_amount_paid*.

data::= double-long, scaled according to the *price_scale* of *unit_charge_active*.

and

returns double-long, scaled according to the *price_scale* of *unit_charge_active*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.4.3.5 set_total_amount_remaining (data)

Sets the *total_amount_remaining* attribute. The value shall be not less than 0. The amount previously in *total_amount_remaining* attribute shall be given as the response parameter.

NOTE *total_amount_remaining* is set without affecting *total_amount_paid*.

data::= double-long, scaled according to the *price_scale* of *unit_charge_active*,

and

returns double-long, scaled according to the *price_scale* of *unit_charge_active*.

4.6.5 Token gateway (class_id = 115, version = 0)

4.6.5.1 General

An instance of the “Token gateway” IC implements the Token Carrier Interface.

NOTE A single instance of the “Token gateway” object is instantiated for each “Account” object and hence each supply contract.

Token gateway		0...n	class_id = 115, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. token	(dyn.)	octet-string				x + 0x08
3. token_time	(dyn.)	octet-string				x + 0x10
4. token_description	(dyn.)	array				x + 0x18
5. token_delivery_method	(dyn.)	enum				x + 0x20
6. token_status	(dyn.)	structure				x + 0x28
Specific methods		***m/o***				
1. enter (data)		**m**				x + 0x30

4.6.5.2 Attribute description

4.6.5.2.1 logical_name

Identifies the “Token gateway” object instance. See 6.2.17.

4.6.5.2.2 token

Contains the unprocessed octet string of the most recently received or token for the purpose of capture in a history profile.

4.6.5.2.3 token_time

Contains, for the most recently received and processed token, the time and date at which a received token arrived at the DLMS®/COSEM server.

octet-string, formatted as specified in 4.1.6.1 for *date_time*.

4.6.5.2.4 token_description

Contains a description of the token for the most recently received and processed token which is supplied by the meter’s application program.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE For example this could contain the type of token (credit or engineering), and/or the token origin.

The use of this attribute is to be described in the token specification or project specific companion specifications.

token_description:: = array token_description_element

token_description_element::= octet-string.

4.6.5.2.5 token_delivery_method

Reflects the route by which the last token was received.

enum:

(0) via remote communications,
(1) via local communications,
(2) via manual entry

4.6.5.2.6 token_status

token_status is a structure containing a generic enumeration that shows the status of the last token operation and an optional bit-string that can be associated with the token status giving extra information about the token *status_code*.

NOTE 1 It is expected that the optional bit string content will be documented in a project specific companion specification.

NOTE 2 The exact meaning and criteria for evaluation of the *status_code* values will be slightly different depending on the token protocol and format. For example a token conforming to IEC 62055-41 has very precise criteria and meanings for each of the terms Validation, Authentication and Token result, while other specifications may have differing terms.

token_status::= structure
{
status_code: enum,
data_value: bit-string
}

status_code: enum:

(0) Token format result OK,
(1) Authentication result OK,
(2) Validation result OK,
(3) Token execution result OK,
(4) Token format failure,
(5) Authentication failure,
(6) Validation result failure,
(7) Token execution result failure,
(8) Token received and not yet processed

On receipt of a token the initial state shall be set to 8 while the token is being processed and until another state can be deduced.

The use of the data_value bit-string field is to be defined in project specific companion specifications.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.5.3 Method description

4.6.5.3.1 enter (data)

This method is invoked to transfer a token to the DLMS®/COSEM server in the form of octet-string. If successful it may return the *token_status* attribute.

data::= octet-string, and returns *token_status.*

4.6.5.4 Additional remarks

There are a number of configurable limits that may be held by “Data” objects and relate to the behaviour and processing of tokens and some aspects of the payment metering system. Some of these limits have been defined in this specification, have standard OBIS codes and are described below:

Max credit limit: On receipt of a token the meter’s application process shall check that the addition of the credit token’s value to the associated credit object(s) will not force the *available_credit* in the “Account” object above the limit programmed in the “Max credit limit” object. If the received token is found to force the *available_credit* to be more than this limit then the meter shall reject the token and return the appropriate status.

Max vend limit: On receipt of a token the meter’s application process shall check that the value of the credit token is not more than the limit programed in the “Max vend limit” object. If the received token is found to be more than this limit then the meter shall reject the token and return the appropriate status.

See also 6.2.17.

4.7 Interface classes for setting up data exchange via local ports and modems

4.7.1 IEC local port setup (class_id = 19, version = 1)

4.7.1.1 Overview

This IC allows modelling the configuration of communication ports using the protocols specified in IEC 62056-21:2002. Several ports can be configured.

IEC local port setup		**0...n**	**class_id = 19, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. default_mode	(static)	enum				x + 0x08
3. default_baud	(static)	enum				x + 0x10
4. prop_baud	(static)	enum				x + 0x18
5. response_time	(static)	enum				x + 0x20
6. device_addr	(static)	octet-string				x + 0x28
7. pass_p1	(static)	octet-string				x + 0x30
8. pass_p2	(static)	octet-string				x + 0x38
9. pass_w5	(static)	octet-string				x + 0x40
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.1.2 Attribute description

4.7.1.2.1 logical_name

Identifies the “IEC local port setup” object instance. See 6.2.18.

4.7.1.2.2 default_mode

Defines the protocol used by the meter on the port.

enum:

(0) protocol according to IEC 62056-21:2002 (modes A…E),
(1) protocol according to IEC 62056-46:2002/AMD1:2006. Using this enumeration value all other attributes of this IC are not applicable,
(2) protocol not specified. Using this enumeration value, attribute 4), prop_baud is used for setting the communication speed on the port. All other attributes are not applicable.

4.7.1.2.3 default_baud

Defines the baud rate for the opening sequence.

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,
(9) 115 200 baud

4.7.1.2.4 prop_baud

Defines the baud rate to be proposed by the meter.

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,
(9) 115 200 baud

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.1.2.5 response_time

Defines the minimum time between the reception of a request (end of request telegram) and the transmission of the response (begin of response telegram).

enum:

(0) 20 ms,
(1) 200 ms

4.7.1.2.6 device_addr

Device address according to IEC 62056-21:2002.

4.7.1.2.7 pass_p1

Password 1 according to IEC 62056-21:2002.

4.7.1.2.8 pass_p2

Password 2 according to IEC 62056-21:2002.

4.7.1.2.9 pass_w5

Password W5 reserved for national applications.

4.7.2 IEC HDLC setup (class_id = 23, version = 1)

4.7.2.1 Overview

This IC allows modelling and configuring communication channels according to IEC 62056-46:2002/AMD1:2006. Several communication channels can be configured.

IEC HDLC setup		**0...n**	**class_id = 23, version = 1**			
Attributes		***Data type***	**Min.**	**Max.**	**Def.**	**Short name**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	9	5	x + 0x08
3. window_size_transmit	(static)	unsigned	1	7	1	x + 0x10
4. window_size_receive	(static)	unsigned	1	7	1	x + 0x18
5. max_info_field_length_transmit	(static)	long-unsigned	32	2030	128	x + 0x20
6. max_info_field_length_receive	(static)	long-unsigned	32	2030	128	x + 0x28
7. inter_octet_time_out	(static)	long-unsigned	20	6000	25	x + 0x30
8. inactivity_time_out	(static)	long-unsigned	0		120	x + 0x38
9. device_address	(static)	long-unsigned	0x0010	0x3FFD		x + 0x40
Specific methods		***m/o***				

NOTE 1 The maximum value of the attributes *max_info_field_length_transmit* and *max_info_field_length_receive* has been increased from 128 to 2 030 for efficiency reasons.

NOTE 2 A *max_info_field_length_receive* of 128 bytes is needed to ensure a minimal performance.

NOTE 3 The maximum value of the *inter-octet-time-out* attribute has been increased from 1 000 ms to 6 000 ms in order to allow using communication media, where long delays may occur. The default value has been changed to 25 ms to align with 6.4.4.3.4 of IEC 62056-46:2002/AMD1:2006.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.2.2 Attribute description

4.7.2.2.1 logical_name

Identifies the “IEC HDLC setup” object instance. See 6.2.20.

4.7.2.2.2 comm_speed

The communication speed supported by the corresponding port.

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,
(9) 115 200 baud

This communication speed can be overridden if the HDLC mode of a device is entered through a special mode of another protocol.

4.7.2.2.3 window_size_transmit

The maximum number of frames that a device or system can transmit before it needs to receive an acknowledgement from a corresponding station. During logon, other values can be negotiated.

4.7.2.2.4 window_size_receive

The maximum number of frames that a device or system can receive before it needs to transmit an acknowledgement to the corresponding station. During logon, other values can be negotiated.

4.7.2.2.5 max_info_length_transmit

The maximum information field length that a device can transmit. During logon, a smaller value can be negotiated.

4.7.2.2.6 max_info_length_ receive

The maximum information field length that a device can receive. During logon, a smaller value can be negotiated.

4.7.2.2.7 inter_octet_time_out

Defines the time, expressed in milliseconds, over which, when no character is received from the primary station, the device will treat the already received data as a complete frame.

4.7.2.2.8 inactivity_time_out

Defines the time, expressed in seconds over which, when no frame is received from the primary station, the device will process a disconnection.

When this value is set to 0, this means that the inactivity_time_out is not operational.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.2.2.9 device_address

Contains the physical device address of a device.

In the case of one byte addressing:

0x00	NO_STATION Address,
0x01…0x0F	Reserved for future use,
0x10…0x7D	Usable address space,
0x7E	'CALLING' device address,
0x7F	Broadcast address

In the case of two byte addressing:

0x0000	NO_STATION address,
0x0001…0x000F	Reserved for future use,
0x0010…0x3FFD	Usable address space,
0x3FFE	'CALLING' physical device address,
0x3FFF	Broadcast address

4.7.3 IEC twisted pair (1) setup (class_id = 24, version = 1)

4.7.3.1 General

Instances of this IC allow setting up data exchange over the medium *twisted pair with carrier signalling* as specified in IEC 62056-3-1:2013. Several communication channels can be configured.

The communication medium *twisted pair with carrier signalling* is widely used in metering. The main advantages of using this medium are the ease of installation and the reliability of communications due to carrier signalling. This medium can be used:

- between Local Network Access Points (LNAPs) and metering end devices (M interface);
- between Local Network Access Points (LNAPs) and Neighbourhood Network Access Points (NNAPs); and
- for direct connection between a HHU and the metering end device.

IEC 62056-3-1:2013 specifies three communication profiles using the medium twisted pair with carrier signalling:

- without DLMS®;
- with DLMS®; and
- with DLMS®/COSEM.

IEC 62056-31:1999 supports only the first two profiles.

The new, DLMS®/COSEM profile introduces a Support Manager Layer entity performing the initialisation of the bus, discovery management, alarm management and communication speed negotiation. It also allows higher baud rates up to 9 600 Bd. The Transport Layer supports segmentation and reassembly.

The IC "IEC Twisted pair (1) set up" (class_id = 24, version = 0) supports the first two communication profiles specified in IEC 62056-31:1999.

The new version 1 supports the DLMS®/COSEM profile. With its introduction, the use of version 0 is deprecated.

The use of the communication profiles specified in IEC 62056-3-1:2013 requires using the registration services provided by the Euridis Association: www.euridis.org.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The following COSEM interface objects are necessary to set up data exchange over the medium *Twisted pair with carrier signalling:*

- “IEC Twisted pair (1) setup”: class_id = 24, version = 1;
- “MAC address”: class_id = 43, version = 0;
- “Data”: class_id = 1, version = 0.

For OBIS codes, see 6.2.21.

4.7.3.2 Overview

Instances of this IC allow setting up data exchange over the medium *twisted pair with carrier signalling* as specified in IEC 62056-3-1:2021. Several communication channels can be configured.

IEC twisted pair (1) setup		**0...n**	**class_id = 24, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum	0		1	x + 0x08
3. comm_speed	(static)	enum	(2)	(7)	(2)	x + 0x10
4. primary_address_list	(static)	primary_address_list_type				x + 0x18
5. tabi_list	(static)	tabi_list_type				x + 0x20
Specific methods		***m/o***				

4.7.3.3 Attribute description

4.7.3.3.1 logical_name

Identifies the “IEC twisted pair setup” object instance. See 6.2.21.

4.7.3.3.2 mode

This attribute specifies the working mode of this interface

enum:

(0) inactive. The interface ignores all frames received,
(1) always active,
(2)... (127) reserved,
(128)...(250) manufacturer specific.

4.7.3.3.3 comm_speed

Holds the communication speed supported by the port.

enum:

(2) 1200 baud,
(3) 2400 baud,
(4) 4800 baud,
(5) 9600 baud,
(6) 19200 baud,
(7) 38400 baud

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE IEC 62056-3-1:2013 supports baud rates from 1 200 to 9 600.

4.7.3.3.4 primary_address_list

Holds the list of Primary Station Addresses (ADP) for which each logical device of the real equipment (the secondary station) has been programmed. See IEC 62056-3-1:2013, 5.2.4.

primary_address_list_type::= array unsigned

4.7.3.3.5 tabi_list

Represents the list of the TAB(i) for which the real equipment (the secondary station) has been programmed in the case of forgotten station call (see IEC 62056-3-1:2013).

When using IEC 62056-3-1 profile with DLMS®/COSEM, the tabi_list attribute is made of only one element of value 0, the value used for the discovery process.

tabi_list_type::= array tabi_element

tabi_element: integer

4.7.3.4 MAC address

Secondary stations – typically metering end devices – hold a secondary station address (ADS). The length of the ADS shall be 6 octets and consists of the elements shown in Table 36. This address shall be worldwide unique and it shall be assigned by the manufacturer to the secondary station. The ADS assigned is valid through the lifetime of the secondary station.

Table 36 – ADS address elements

Elements of ADS	Description	Length (byte)	Type	Range
manufacturer_id	Assigned upon request by the Euridis Association to the manufacturer. It shall be used in all devices using the *Twisted pair with carrier signalling* medium and made by this manufacturer.	1	BCD	0-99
year_of_manufacture	Holds the year of manufacturing the device (the lower two numbers only).	1		0-99
equipment_id	Related to the type of equipment concerned and provides information on the functionality available. Like the manufacturer_id, it is assigned by the Euridis Association.	1		0-99
device_serial_number	The manufacturing number of the device assigned by the manufacturer. It should have the same value as the value held by the object Device ID 1, seeIEC 62056-6-1:2023, Table 8.	3		000001-999999 000000 is reserved
EXAMPLE 031267123456: 03: ITRON International; 12: Year 2012; 67: Smart meter LINKY Single phase 123456: Serial number beginning each year from 000001.				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.3.5 Fatal error register

Each device implementing the DLMS®/COSEM communication profile specified in IEC 62056-3-1:2013 shall provide an error register holding the result of the last communication with the primary station. The structure of the fatal error register shall be as specified in Table 37.

Table 37 – Fatal error register

Ref	Name	Description
Bit 0	EP-3F	Transmission error. The time out TOE is elapsed without the byte being sent, leading to a non-ability to send the remaining part of the frame.
Bit 1	EP-4F	Reception error. The number of bytes received is higher than the maximum expected.
Bit 2	EP-5F	Expiry of TARSO wake-up while receiving an RSO frame. Not relevant for secondary station (server); concerns the primary station (client) only.
Bit 3	EL-1F	Alarm indication received during an association. No relevant. Concern the primary station (client) only.
Bit 4	EL-2F	Incorrect response from the Secondary Station after MaxRetry repeated transmissions of a request.
Bit 5	EA-1F	Incorrect TAB from the server. Not relevant for secondary station (server); concerns the primary station (client) only.
Bit 6	EA-2F	Authentication error on the data received from the server. Not relevant for secondary station (server); concerns the primary station (client) only.
Bit 7	EA-3F	Authentication error detected by the secondary station.

4.7.4 Modem configuration (class_id = 27, version = 1)

4.7.4.1 Overview

This IC allows modelling the configuration and initialisation of modems used for data transfer from/to a device. Several modems can be configured.

Modem configuration		**0...n**	**class_id = 27, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	9	5	x + 0x08
3. initialization_string	(static)	array				x + 0x10
4. modem_profile	(static)	array				x + 0x18
Specific methods		***m/o***				

4.7.4.2 Attribute description

4.7.4.2.1 logical_name

Identifies the "Modem configuration" object instance. See 6.2.6.

4.7.4.2.2 comm_speed

The communication speed between the device and the modem, not necessarily the communication speed on the WAN.

enum:

300 baud,
600 baud,
1 200 baud,
2 400 baud,
4 800 baud,
9 600 baud,
19 200 baud,
38 400 baud,
57 600 baud,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

115 200 baud

4.7.4.2.3 initialization_string

Contains all the necessary initialization commands to be sent to the modem in order to configure it properly. This may include the configuration of special modem features.

```
array        initialization_string_element

initialization_string_element::= structure
{
    request:                    octet-string,
    response:                   octet-string,
    delay_after_response:       long-unsigned
}
```

If the array contains more than one initialization_string_element, the requests are sent in a sequence. The next request is sent after the expected response matching the previous request and waiting a delay_after_response time [ms], to allow the modem to execute the request.

NOTE It is assumed that the modem is pre-configured so that it accepts the initialization_string. If no initialization is needed, the initialization string is empty.

4.7.4.2.4 modem_profile

Defines the mapping from Hayes standard commands/responses to modem specific strings.

```
array        modem_profile_element

modem_profile_element: octet-string
```

The modem_profile array shall contain the corresponding strings for the modem used in following order:

```
Element 0:      OK,
Element 1:      CONNECT,
Element 2:      RING,
Element 3:      NO CARRIER,
Element 4:      ERROR,
Element 5:      CONNECT 1 200,
Element 6       NO DIAL TONE,
Element 7:      BUSY,
Element 8:      NO ANSWER,
Element 9:      CONNECT 600,
Element 10:     CONNECT 2 400,
Element 11:     CONNECT 4 800,
Element 12      CONNECT 9 600,
Element 13:     CONNECT 14 400,
Element 14:     CONNECT 28 800,
Element 15:     CONNECT 33 600,
Element 16:     CONNECT 56 000
```

4.7.5 Auto answer (class_id = 28, version = 2)

4.7.5.1 General

NOTE 1 Version 1 of the Auto answer class was an interim version.

Version 0 of the Auto answer class models how the device handles incoming calls to request the connection of the modem.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

In version 2, new capabilities are added to manage wake-up requests that may be in the form of a wake-up call or a wake-up message e.g. an (empty) SMS message. After a successful wake-up request, the device connects to the network. See also Annex A.

For both functions, additional security is provided by adding the possibility of checking the calling number: calls or messages are accepted only from a pre-defined list of callers. This feature requires the presence of a calling line identification (CLI) service in the communication network used.

NOTE 2 The wake-up process is fully decoupled from AL services, i.e. a wake-up message cannot contain any xDLMS service requests. This is to avoid creating a backdoor. xDLMS messages may be exchanged in SMS messages once the wake-up process is completed.

Auto answer		**0...n**	**class_id = 28, version = 2**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. listening_window	(static)	array				x + 0x10
4. status	(dyn.)	enum				x + 0x18
5. number_of_calls	(static)	unsigned				x + 0x20
6. number_of_rings	(static)	nr_rings_type				x + 0x28
7. list_of_allowed_callers	(static)	array				x + 0x30
Specific methods		***m/o***				

4.7.5.2 Attribute description

4.7.5.2.1 logical_name

Identifies the “Auto answer” object instance. See 6.2.6.

4.7.5.2.2 mode

Defines the working mode of the line when the device is auto answering.

enum:

(0) line dedicated to the device,
(1) shared line management with a limited number of calls allowed. Once the number of calls is reached, the window status becomes inactive until the next start date, whatever the result of the call,
(2) shared line management with a limited number of successful calls allowed. Once the number of successful communications is reached, the window status becomes inactive until the next start date,
(3) currently no modem connected,
(200...255) manufacturer specific modes

4.7.5.2.3 listening_window

Defines the time points when the communication window(s) become active (start_time) and inactive (end_time). The start_time implicitly defines the period.

EXAMPLE When the day of month is not specified (equal to 0xFF) this means that we have a daily listening window management. Daily, monthly ...window management can be defined.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
array        window_element

window_element::= structure
{
        start_time:  octet-string,
        end_time:    octet-string
}
```

start_time and end_time are formatted as specified in 4.1.6.1 for *date-time*.

4.7.5.2.4 status

Here the status of the window is defined.

enum:

(0) Inactive: the device will manage no new incoming call. This status is automatically reset to Active when the next listening window starts,
(1) Active: the device can answer to the next incoming call,
(2) Locked: This value can be set automatically by the device or by a specific client when this client has completed its reading session and wants to give the line back to the customer before the end of the window duration. This status is automatically reset to Active when the next listening window starts.

4.7.5.2.5 number_of_calls

This number is the reference used in modes 1 and 2.

When set to 0, this means there is no limit.

4.7.5.2.6 number_of_rings

Defines the number of rings before the meter connects the modem. Two cases are distinguished: The number of rings within the window defined by the attribute *listening_window* and the number of rings outside the *listening_window*.

```
nr_rings_type::= structure
{
        nr_rings_in_window:      unsigned (0 = no connection in the      window),
        nr_rings_out_of_window: unsigned (0 = no connection outside the window)
}
```

If the number of rings inside and outside the window is the same, the modem always connects regardless of the settings of the *listening_window*.

4.7.5.2.7 list_of_allowed_callers

Contains an – optional – list of calling numbers which further limits the connectivity of the modem based on the calling number. It also controls the acceptance of wake-up calls or wake-up messages (e.g. SMS) from a calling number.

This requires the presence of a calling line identification (CLI) service in the communication network used.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
list_of_allowed_callers::= array list_of_allowed_callers_element

list_of_allowed_callers_element::= structure
{
      caller_id:   octet-string,
      call_type:   enum
}
```

– the caller_id element holds a calling number from which calls or messages (e.g. SMS) are accepted. The wild-card characters '?' and '*' are supported. With '?' any single character matches, with '*' any character string matches. '*' can only be used at the beginning or at the end of a number, but neither in between nor alone.

EXAMPLE 1: "+994193500" = only calls from "+994193500" are accepted.

EXAMPLE 2: "+9941935????" = calls from all numbers in the range of "+99419350000" to "+99419359999" are accepted.

EXAMPLE 3: "7777*" = calls from all numbers starting with "7777" are accepted.

EXAMPLE 4: "*9000" = calls from all numbers ending with "9000" are accepted.

– the call_type element defines the purpose of the call, i.e. if it's a standard CSD call or a wake-up call / wake-up message.

enum:

(0) = normal CSD call; the modem only connects if the calling number matches an entry in the list. This is tested in addition to all other attributes, e.g. *number_of_rings*, *listening_windows*, etc.

(1) = wake-up request; calls or messages from this calling number are handled as wake-up requests. The wake-up request is processed immediately regardless of all other attributes like *number_of_rings* and *listening_window* (except if the calling number is also present in the list of normal CSD calls, see below).

The received message shall be completely empty; otherwise it is not treated as a wake-up message.

If the message contains a valid xDLMS APDU from a client in a pre-established AA, the corresponding xDLMS service is executed instead of processing the wake-up request.

If the message is not empty but does not contain any valid xDLMS APDU from a client in a pre-established AA then the server does not react.

For a call of type (1) the modem *does not* connect – independently of the outcome of the wake-up process.

For a call of both type (1) and type (0): see below.

If the same calling number is defined as initiator of a normal CSD call (type (0)) and as initiator of a wake-up request (type (1)) the following rule applies for the incoming calls: the wake-up request is only processed if the caller disconnects the line before the *number_of_rings* criterion (depending on *listening _window*) is reached. If the *number_of_rings criterion* is met then a modem connection is established.

The *number_of_rings* parameter shall be set large enough to allow the initiator of the call to control the behaviour of the receiver of the call without any knowledge of the time instances of the rings at the receiver's side.

NOTE If the *list_of_allowed_callers* is empty (= array [0]) the auto answer function operates in type (0) = normal CSD call and the modem connects independently of the calling number.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.6 Auto connect (class_id = 29, version = 2)

4.7.6.1 Overview

Version 1 of the “Auto connect” class models how the device performs auto dialling or sends messages using various services.

In version 2 new capabilities are added to model the connection of the device to a communication network. Network connection may be permanent, within a time window or on invocation of the connect method.

Auto connect		**0...n**	**class_id = 29, version = 2**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. repetitions	(static)	unsigned				x + 0x10
4. repetition_delay	(static)	long-unsigned				x + 0x18
5. calling_window	(static)	array				x + 0x20
6. destination_list	(static)	array				x + 0x28
Specific methods		***m/o***				
1. connect (data)		o				x + 0x30

4.7.6.2 Attribute description

4.7.6.2.1 logical_name

Identifies the “Auto connect” object instance. See 6.2.6.

4.7.6.2.2 mode

Controls the auto connect functionality in terms of the timing, the message type and the infrastructure to be used.

Modes (1) to (3) are dedicated to CSD services.

Modes (4) to (6) are dedicated to sending specific messages using a specific infrastructure.

Modes (101) to (104) apply to packet switched network connections only (e.g. GPRS).

enum:

(0) no auto connect; the device never connects,
(1) auto dialling allowed anytime, the values defined in the *calling_window* are ignored,
(2) auto dialling allowed within the validity time of the *calling_window*,
(3) “regular” auto dialling allowed within the validity time of the *calling_window*; “alarm” initiated auto dialling allowed anytime,
(4) SMS sending via Public Land Mobile Network (PLMN),
(5) SMS sending via PSTN,
(6) email sending,
(7…99) reserved,
(101) the device is permanently connected to the communication network,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(102) the device is permanently connected to the communication network within the validity time of the calling window. The device is disconnected outside the calling window. No connection possible outside the calling window,
(103) the device is permanently connected to the communication network within the validity time of the calling window. The device is disconnected outside the calling window but it connects to the communication network as soon as the connect method is invoked,
(104) the device is usually disconnected. It connects to the communication network as soon as the connect method is invoked,
(105…199) reserved,
(200…255) manufacturer specific modes.

4.7.6.2.3 repetitions

The maximum number of retries in case of unsuccessful connection attempts.

4.7.6.2.4 repetition_delay

The time delay, expressed in seconds until an unsuccessful connection attempt can be repeated.

repetition_delay = 0 means delay is not specified.

4.7.6.2.5 calling_window

Contains the time points when the window becomes active (start_time), and inactive (end_time). The start_time implicitly defines the period.

EXAMPLE When the day of month is not specified (equal to 0xFF) this means that the calling window is managed on a daily basis. Daily, monthly …window management can be defined.

```
array        window_element

window_element::= structure
{
        start_time:  octet-string,
        end_time:    octet-string
}
```

start_time and end_time are formatted as specified in 4.1.6.1 for *date-time.*

4.7.6.2.6 destination_list

Contains the list of destinations (for example phone numbers, email addresses or their combinations) where the message(s) have to be sent under certain conditions. The conditions and their link to the elements of the array are not defined here.

```
array        destination

destination::= octet-string
```

4.7.6.3 Method description

4.7.6.4 connect (data)

Initiates the connection process to the communication network according to the rules defined via the mode attribute.

```
data::= integer (0)
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.7 GPRS modem setup (class_id = 45, version = 0)

4.7.7.1 Overview

This IC allows setting up GPRS modems, by handling all data necessary data for modem management.

GPRS modem setup		0...n	class_id = 45, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. APN	(static)	octet-string				x + 0x08
3. PIN_code	(static)	long-unsigned				x + 0x10
4. quality_of_service	(static)	structure				x + 0x18
Specific methods		***m/o***				

4.7.7.2 Attribute description

4.7.7.2.1 logical_name

Identifies the “GPRS modem setup” object instance. See 6.2.23.

4.7.7.2.2 APN

Defines the access point name of the network.

4.7.7.2.3 PIN_code

Holds the personal identification number.

4.7.7.2.4 quality_of_service

Specifies the quality of service parameters. It is a structure of 2 elements:

– the first element defines the default or minimum characteristics of the network concerned. These parameters have to be set to best effort value;
– the second element defines the requested parameters.

```
quality_of_service::= structure
{
        default:            qos_element,
        requested:          qos_element
}
qos_element::= structure
{
        precedence:             unsigned,
        delay:                  unsigned,
        reliability:            unsigned,
        peak throughput:        unsigned,
        mean throughput:        unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.8 GSM diagnostic (class_id: 47, version: 2)

4.7.8.1 Overview

The cellular network is undergoing constant changes in terms of registration status, signal quality, etc. It is necessary to monitor and log the relevant parameters in order to obtain diagnostic information that allows identifying communication problems in the network.

An instance of the “GSM diagnostic” class stores parameters of the GSM/GPRS, UMTS, CDMA or LTE network necessary for analysing the operation of the network.

A GSM diagnostic “Profile generic” object is also available to capture the attributes of the GSM diagnostic object, see IEC 62056-6-1:2023, 6.5.

GSM diagnostic		**0...n**	**class_id = 47, version = 2**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. operator	(dyn.)	visible-string				x + 0x08
3. status	(dyn.)	enum	0	255	0	x + 0x10
4. cs_attachment	(dyn.)	enum	0	255	0	x + 0x18
5. ps_status	(dyn)	enum	0	255	0	x + 0x20
6. cell_info	(dyn.)	cell_info_type				x + 0x30
7. adjacent_cells	(dyn.)	array				x + 0x38
8. capture_time	(dyn.)	date-time				x + 0x40
Specific methods		***m/o***				

4.7.8.2 Attribute description

4.7.8.2.1 logical_name

Identifies the “GSM Diagnostic” object instance. For logical names, see 6.2.23.

4.7.8.2.2 operator

Holds the name of the network operator e.g. “YourNetOp”

4.7.8.2.3 status

Indicates the registration status of the modem.

enum:

(0) not registered,
(1) registered, home network,
(2) not registered, but MT is currently searching a new operator to register to,
(3) registration denied,
(4) unknown,
(5) registered, roaming,
(6)... (255) reserved

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.8.2.4 cs_attachment

Indicates the current circuit switched status.

enum:

(0) inactive,
(1) incoming call,
(2) active,
(3)...(255) reserved

4.7.8.2.5 ps_status

The *ps_status* value field indicates the packet switched status of the modem.

enum:

(0) inactive,
(1) GPRS,
(2) EDGE,
(3) UMTS,
(4) HSDPA,
(5) LTE,
(6) CDMA,
(7) LTE Cat M1,
(8) LTE Cat NB1,
(9) LTE Cat NB2,
(10)...(255) reserved

4.7.8.2.6 cell_info

Represents the cell information:

```
cell_info_type::= structure
{
    cell_ID:            double-long-unsigned,
    location_ID:        long-unsigned,
    signal_strength:    unsigned,
    ber:                unsigned,
    mcc:                long-unsigned,
    mnc:                long-unsigned,
    channel_number:     double-long-unsigned
}
```

Where:

– cell_ID: Four-byte cell ID in hexadecimal format;

– location_ID: Two-byte location area code (LAC) in the case of GSM networks or Tracking Area Code (TAC) in the case of UMTS, CDMA or LTE networks in hexadecimal format (e.g. "00C3" equals 195 in decimal);

– signal_strength: Represents the signal quality:

(0) –113 dBm or less,
(1) –111 dBm,
(2...30) –109...-53 dBm,
(31) –51 or greater,
(99) not known or not detectable;

– ber: Bit Error Rate (BER) measurement in percent:

(0...7) as RXQUAL_n values specified in ETSI GSM 05.08:1996, 8.2.4
(99) not known or not detectable.

– mcc: Mobile Country Code of the serving network, as defined in ITU-T E.212 (05.2008);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– mnc: Mobile Network Code of the serving network, as defined in ITU-T E.212 (05.2008);

– channel_number: Represents the absolute radio-frequency channel number (ARFCN or EARFCN for LTE network).

4.7.8.2.7 adjacent_cells

```
array              adjacent_cell_info

adjacent_cell_info::= structure
{
    cell_ID:          double-long-unsigned,
    signal_strength:  unsigned
}
```

Where:

– cell_ID: Four-byte cell ID in hexadecimal format;

– signal_strength: Represents the signal quality:

(0) –113 dBm or less,
(1) –111 dBm,
(2...30) –109...-53 dBm,
(31) –51 or greater,
(99) not known or not detectable.

4.7.8.2.8 capture_time

Holds the date and time when the data have been last captured.

date-time is formatted as specified in 4.1.6.1.

4.7.9 LTE monitoring (class_id: 151, version: 1)

4.7.9.1 Overview

Instances of the 'LTE monitoring' IC allow monitoring LTE modems by handling all data necessary data for this purpose.

Version 1 of the LTE monitoring IC also covers LTE Cat M1 and NB-IoT networks.

LTE monitoring		**0...n**	**class_id = 151, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. LTE_network_parameters	(dyn.)	LTE_network_ parameters_type				x + 0x08
3. LTE_quality_of_service	(dyn)	LTE_QoS_type				x + 0x10
Specific methods		***m/o***				

4.7.9.2 Attribute description

4.7.9.2.1 logical_name

Identifies the "LTE monitoring" object instance. See 6.2.23.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.9.2.2 LTE_network_parameters

Represents the network parameters for the LTE network

```
LTE_network_parameters_type::= structure

{
        T3402:                  long-unsigned,
        T3412:                  long-unsigned,
        T3412ext2:              double-long-unsigned,
        T3324                   long-unsigned,
        TeDRX                   double-long unsigned,
        TPTW                    long-unsigned,
        qRxlevMin:              integer,
        qRxlevMinCE-r13:        integer,
        qRxlevMinCE1-r13        integer
}
```

Where:

– T3402 timer in seconds, used on PLMN selection procedure and sent by the network to the modem. Refer to 3GPP TS 24.301 V13.11.0 (2018-01) for details;

– T3412 timer in seconds, used to manage the periodic tracking area updating procedure and sent by the network to the modem. Refer to 3GPP TS 24.301 V13.11.0 (2018-01) for details;

– T3412ext2 timer in seconds (extended periodic tracking area update timer). Refer to 3GPP TS 24.301 V13.11.0 (2018-01) and 3GPP TS 24.008 V13.7.0 (2016-10) for details;

– T3324 timer in seconds (Power saving mode active timer). Refer to 3GPP TS 24.301 V13.11.0 (2018-01) and 3GPP TS 24.008 V13.7.0 (2016-10) for details;

– TeDRX timer (Extended Idle mode DRX cycle timer). Refer to 3GPP TS 24.301 V13.11.0 (2018-01) and 3GPP TS 24.008 V13.7.0 (2016-10) for details;

– The double-long unsigned value shall be multiplied by 0,01 to get the real value in seconds. E.g. 512 represents 5,12 seconds;

– TPTW timer (Extended Idle mode DRX paging time window). Refer to 3GPP TS 24.301 V13.11.0 (2018-01) and 3GPP TS 24.008 V13.7.0 (2016-10) for details;

– The long unsigned value shall be multiplied by 0,01 to get the real value in seconds. E.g. 512 represents 5,12 seconds;

– qRxlevMin the minimum required Rx level in the cell in dBm as defined in 3GPP TS 36.304 V13.8.0 (2018-01);

– qRxlevMinCE-r13 the minimum required Rx level in enhanced coverage CE Mode A, LTE Cat M1. For this mode a value from -70…-22 in steps of 1 is required;

NOTE 1 This field is not used in case of LTE Cat NB1 or NB2.

– qRxlevMinCE1-r13 the minimum required Rx level in enhanced coverage CE Mode B, LTE Cat M1. For this mode a value from -78…-22 in steps of 1.

NOTE 2 This field is not used in case of LTE Cat NB1 orNB2.

4.7.9.2.3 LTE_quality_of_service

Represents the quality of service of the LTE network

```
LTE_QoS_type::= structure
{
        (N)RSRQ:                integer,
        (N)RSRP:                integer,
        SNR:                    integer,
        Coverage Enhancement:   enum
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

(N)RSRQ represents the signal quality as defined in 3GPP TS 36.133. The use of this parameter can be determined from the GSM diagnostic IC, ps_status attribute:

- For LTE Cat M1, a value range from -30 up to 46 is necessary to represent RSRQ. Refer to 3GPP TS 36.133 V13.11.0 (2018-04) for details.
- For LTE Cat NB1 and LTE Cat NB2 a value range from -30 up to 46 is necessary to represent NRSRQ. Refer to 3GPP TS 36.133 V14.4.0 (2017-07) for details.

(-30)	-34 dB,
(-29)	-33,5 dB,
(-28)..(-1)	-33 dB .. -19,5 dB,
(0)	-19,5 dB or less,
(1)	-19 dB,
(2)..(31)	-18,5 dB .. -4 dB,
(32)	-3.5 dB,
(33)	-3 dB,
(34)	-3 dB or less,
(35)	-2.5 dB,
(36)	-2 dB,
(37)..(45)	-1.5 dB .+2.5 dB,
(46)	2.5 dB or better,
(99)	not known or not detectable,
other values	reserved

(N)RSRP represents the signal level as defined in 3GPP TS 36.133. The use of this parameter is determined from the GSM diagnostic IC, ps_status attribute:

- For ps_status = 7 (LTE Cat M1) a value range from -17 up to 97 is necessary to represent RSRP. Refer to 3GPP TS 36.133 V13.11.0 (2018-04) for details

(-17)	-156 dBm,
(-16)	-155 dBm,
(-15)..(-1)	-154 dBm .. -140 dBm,
(0)	-140 dBm or less,
(1)	-139 dB,
(2)	-138 dBm,
(3)..(96)	-137 dBm .. -44 dBm,
(97)	-44 dBm or better,
(127)	not known or not detectable,
other values	reserved

NOTE 1 Depending on the use of Coverage Enhancement modes, a reported value of (0) will be used or not (for devices that do not use CE modes, values (-17..-1) will not be used. A Reported value of (0) means in that case a RSRP < -140 dBm).

- For ps_status = 8 or 9 (LTE Cat NB1 or LTE Cat NB2) a value range from 0 to 113 is necessary to represent NRSRP value. Refer to 3GPP TS 36.133 V14.4.0 (2017-07) for details.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(0)	-156 dBm,
(1)	-155 dBm,
(2)	-154 dBm
(3)..(112)	-153 dBm .. -44 dBm,
(113)	-44 dBm or better,
(127)	not known or not detectable,
other values	reserved

NOTE 2 For the NRSRP value (NB-IoT) the mapping is different from the mapping for the RSRP value (LTE Cat M1).

- SNR the Signal to Noise Ratio in a range from -20 dB to 50 dB. Refer to 3GPP TS 36.101 V15.4.0 (2019-01) for details.
- Coverage Enhancement Coverage Enhancement Mode A or B in case of LTE Cat M1, CE Level 0,1 or 2 in case of LTE Cat NB1/NB2. Refer to 3GPP TS 36.331 V15.5.1 (2019-05), 3GPP TS 36.321 V15.5.0 (2019-05) and 3GPP TS 36.213 V15.5.0 (2019-05) for details.

Enum	LTE Cat M1	NB-IoT
(0)	CE Mode A	CE Level 0
(1)	CE Mode B	CE Level 1
(2)	(not used)	CE Level 2
other values	reserved	

4.8 Interface classes for setting up data exchange via M-Bus

4.8.1 Overview

The M-Bus related interface classes specified in this subclause are used in two different scenarios:

a) a DLMS®/COSEM server hosted by a M-Bus master and exchanging dedicated M-Bus APDUs with M-Bus slaves;

b) a DLMS®/COSEM client hosted by a M-Bus master and exchanging DLMS®/COSEM APDUs with DLMS®/COSEM servers hosted by M-Bus slaves.

In case a) instances of the following M-Bus interface classes are used to set up and manage the M-Bus media in the DLMS®/COSEM server:

- M-Bus client (class_id = 72), see 4.8.3;
- M-Bus master port setup (class_id = 74), see 4.8.5;
- M-Bus diagnostic (class_id = 77, version = 0), see 4.8.7.

In case b) instances of the following M-Bus interface classes are used in the DLMS®/COSEM server:

- DLMS®/COSEM server M-Bus port setup (class_id = 76), see 4.8.6;
- M-Bus slave port setup (class_id = 25), see 4.8.2; and/or
- Wireless Mode Q channel (class_id = 73), see 4.8.4;
- M-Bus diagnostic (class_id = 77, version = 0), see 4.8.7.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.2 M-Bus slave port setup (class_id = 25, version = 0)

4.8.2.1 Overview

NOTE The name of this IC has been changed from “M-BUS port setup” to “M-Bus slave port setup”, to indicate that it serves to set up data exchange when a COSEM server communicates with a COSEM client using wired M-Bus.

This IC allows modelling and configuring communication channels according to EN 13757-2:2004. Several communication channels can be configured.

M-Bus slave port setup		**0...n**	**class_id = 25, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. default_baud	(static)	enum	0	5	0	x + 0x08
3. avail_baud	(static)	enum	0	7		x + 0x10
4. addr_state	(static)	enum				x + 0x18
5. bus_address	(static)	unsigned				x + 0x20
Specific methods		***m/o***				

4.8.2.2 Attribute description

4.8.2.2.1 logical_name

Identifies the “M-Bus slave port setup” object instance. See 6.2.22.

4.8.2.2.2 default_baud

Defines the baud rate for the opening sequence.

enum:

(0) 300 baud,
(3) 2 400 baud,
(5) 9 600 baud

4.8.2.2.3 avail_baud

Defines the baud rates that can be negotiated after startup.

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.2.2.4 addr_state

Defines whether or not the device has been assigned an address since last power up of the device.

enum:

(0) Not assigned an address yet,
(1) Assigned an address either by manual setting, or by automated method.

4.8.2.2.5 bus_address

The currently assigned address on the bus for the device.

NOTE If no bus address is assigned, the value is 0.

4.8.3 M-Bus client (class_id = 72, version = 1)

4.8.3.1 Overview

Instances of the “M-Bus client” allow setting up M-Bus slave devices using wired M-Bus and to exchange data with them. Each “M-Bus client” object controls one M-Bus slave device. For details on the M-Bus dedicated application layer, see EN 13757-3:2013.

NOTE Version 1 of the “M-Bus client” IC is in line with EN 13757-3:2013.

The M-Bus client device may have one or more physical M-Bus interfaces, which can be configured using instances of the “M-Bus master port setup” IC, see 4.8.5.

An M-Bus slave device is identified with its Primary Address, Identification Number, Manufacturer ID etc. as defined in EN 13757-3:2013, Clause 5, Variable Data Send and Variable Data respond. These parameters are carried by the respective attributes of the M-Bus client IC.

Values to be captured from an M-Bus slave device are identified by the *capture_definition* attribute, containing a list of data identifiers (DIB, VIB) for the M-Bus slave device. The data are captured periodically or on an appropriate trigger. Each data element is stored in an M-Bus value object, of IC “Extended register”. M-Bus value objects may be captured in M-Bus “Profile generic” objects, eventually along with other, non M-Bus specific objects.

Using the methods of “M-Bus client” objects, M-Bus slave devices can be installed and de-installed.

It is also possible to send data to M-Bus slave devices and to perform operations like resetting alarms, synchronizing the clock, transferring an encryption key, etc.

Configuration field as defined in EN 13757-3:2013, 5.12 provides information about the encryption mode and number of encrypted bytes.

Encryption key status provides information if encryption key has been set, transferred to M-Bus slave device and is in use with M-Bus slave device.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

M-Bus client		0...n	class_id = 72, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name	(static)	octet-string				x
2. mbus_port_reference	(static)	octet-string				x + 0x08
3. capture_definition	(static)	array			empty	x + 0x10
4. capture_period	(static)	double-long-unsigned			0	x + 0x18
5. primary_address		unsigned			0	x + 0x20
6. identification_number	(dyn.)	double-long-unsigned			0	x + 0x28
7. manufacturer_id	(dyn.)	long-unsigned			0	x + 0x30
8. version	(dyn.)	unsigned			0	x + 0x38
9. device_type	(dyn.)	unsigned			0	x + 0x40
10. access_number	(dyn.)	unsigned			0	x + 0x48
11. status	(dyn.)	unsigned			0	x + 0x50
12. alarm	(dyn.)	unsigned			0	x + 0x58
13. configuration	(dyn.)	long-unsigned			0	x + 0x60
14. encryption_key_status	(dyn.)	enum			0	x + 0x68
Specific methods		***m/o***				
1. slave_install (data)		o				x + 0x70
2. slave_deinstall (data)		o				x + 0x78
3. capture (data)		o				x + 0x80
4. reset_alarm (data)		o				x + 0x88
5. synchronize_clock (data)		o				x + 0x90
6. data_send (data)		o				x + 0x98
7. set_encryption_key (data)		o				x + 0xA0
8. transfer_key (data)		o				x + 0xA8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.3.2 Attribute description

4.8.3.2.1 logical_name

Identifies the “M-Bus client” object instance. See 6.2.22.

4.8.3.2.2 mbus_port_reference

Provides reference to an “M-Bus master port setup” object, used to configure an M-Bus port, each interface allowing to exchange data with one or more M-Bus slave devices.

4.8.3.2.3 capture_definition

Provides the *capture_definition* for M-Bus slave devices.

NOTE 1 This attribute can be pre-configured or written as part of the installation procedure.

```
array        capture_definition_element

capture_definition_element::= structure
{
      data_information_block:      octet-string,
      value_information_block:     octet-string
}
```

NOTE 2 The elements data_information_block and value_information_block correspond to Data Information Block (DIB) and Value Information Block (VIB) described in EN 13757-3:2013, 6.2 and Clause 7 respectively.

4.8.3.2.4 capture_period

>= 1: Automatic capturing assumed. Specifies the capture period in seconds.

0: No automatic capturing: capturing is triggered externally or capture events occur asynchronously.

4.8.3.2.5 primary_address

Carries the primary address of the M-Bus slave device. The range is 0...250.

Each M-bus device is bound to a channel of the M-Bus master. However, there is no direct link between the primary address and the channel number.

NOTE 1 The specification of the B field of the OBIS codes limits the range to 1...64 within one logical device. See IEC 62056-6-1:2023, 5.2.

If the slave device is already configured and thus, its primary address is different from 0, then this value shall be written to the *primary_address* attribute. From this moment, the data exchange with the M-Bus slave device is possible.

Otherwise, the *slave_install* method shall be used; see 4.8.3.3.1.

NOTE 2 The *primary_address* attribute cannot be used to store a desired primary address for an unconfigured slave device. If the *primary_address* attribute is set, this means that the M-Bus client can immediately operate with this primary address, which is not the case with an unconfigured slave device.

4.8.3.2.6 identification_number

Carries the Identification Number element of the data header as specified in EN 13757-3:2013, 5.5.

This attribute, together with attributes 7, 8 and 9 are filled with the values found in the first message received after installation.

If in subsequent messages these values are not the same, the message is discarded.

4.8.3.2.7 manufacturer_id

Carries the Manufacturer Identification element of the data header as specified in EN 13757-3:2013, 5.6.

4.8.3.2.8 version

Carries the Version element of the data header as specified in EN 13757-3:2013, 5.7.

4.8.3.2.9 device_type

Carries the Device type identification element of the data header as specified in EN 13757-3:2013, 5.8, Table 6.

4.8.3.2.10 access_number

Carries the Access Number element of the data header as specified in EN 13757-3:2013, 5.9.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.3.2.11 status

Carries the Status byte element of the data header as specified in EN 13757-3:2013, 5.10, Tables 7 and 8.

It is updated with every readout of the M-Bus slave device.

4.8.3.2.12 alarm

Carries the Alarm state specified in EN 13757-3:2013, Annex D.

It is updated with every readout of the M-Bus slave device.

4.8.3.2.13 configuration

Carries the Configuration field (previously: Signature field) as specified in EN 13757-3:2013, 5.12. It contains information about the encryption mode and the number of encrypted bytes.

It is updated with every readout of the M-Bus slave device.

4.8.3.2.14 encryption_key_status

Provides information on the status of the encryption key. See also Annex B.

enum:

(0) no encryption key,
(1) encryption_key set,
(2) encryption_key transferred,
(3) encryption_key set and transferred,
(4) encryption_key in use.

4.8.3.3 Method description

4.8.3.3.1 slave_install (data)

Installs a slave device, which is yet unconfigured (its primary address is 0).

data::= unsigned

This method can be successfully invoked only if the current value of the *primary_address* attribute is 0. The following actions are performed:

– the M-Bus address 0 is checked for presence of a new device;
– if no uninstalled M-Bus slave is found, the method invocation fails;
– if the *slave_install* method is invoked with “0” as a parameter, then the primary address is assigned automatically. This is done by checking the *primary_address* attribute of all M-Bus client objects in the DLMS®/COSEM device and then selecting the first unused number. The *primary_address* attribute is set to this address and it is then transferred to the M-Bus slave device;
– if the *slave_install* method is invoked with a primary address (other than 0) as a parameter, then the *primary_address* attribute is set to this value and it is then transferred to the M-Bus slave device.

NOTE Unconfigured slave devices are configured with primary address as specified in EN 13757-3:2013, Annex E.5.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.3.3.2 slave_deinstall (data)

De-installs the slave device. The main purpose of this service is to de-install the M-Bus slave device and to prepare the master for the installation of a new device. The following actions are performed:

– the M-Bus address is set to 0 in the M-Bus slave device;
– the encryption key transferred previously to the M-Bus slave device is destroyed; the default key is not affected;
– the *encryption_key_status* is set to (0): no encryption_key;
– the attribute *primary_address* is also set to 0.

NOTE A new M-Bus slave can be installed only once the value of the *primary_address* attribute is 0.

data::= unsigned (0)

4.8.3.3.3 capture (data)

Captures values – as specified by the *capture_definition* attribute – from the M-Bus slave device.

data::= integer (0)

4.8.3.3.4 reset_alarm (data)

Resets alarm state of the M-Bus slave device.

data::= integer (0)

4.8.3.3.5 synchronize_clock (data)

Synchronize the clock of the M-Bus slave device with that of the M-Bus client device.

data::= integer (0)

4.8.4 Wireless Mode Q channel (class_id = 73, version = 1)

4.8.4.1 Overview

Instances of this IC define the operational parameters for communication using the mode Q interfaces. See also EN 13757-5:2015.

Wireless Mode Q channel		**0...n**	**class_id = 73, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. addr_state	(static)	enum				x + 0x08
3. device_address	(static)	octet-string				x + 0x10
4. address_mask	(static)	octet-string				x + 0x18
Specific methods		***m/o***				

4.8.4.2 Attribute description

4.8.4.2.1 logical_name

Identifies the “Wireless Mode Q channel” object instance. See 6.2.22.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.4.2.2 addr_state

Defines whether or not the device has been assigned an address since last power up of the device.

enum:

(0) not assigned an address yet,
(1) assigned an address either by manual setting or by automated method.

4.8.4.2.3 device_address

The currently assigned address of the device on the network.

4.8.4.2.4 address_mask

The group address the device will respond to when short form addressing is used.

4.8.5 M-Bus master port setup (class_id = 74, version = 0)

4.8.5.1 Overview

Instances of this IC define the operational parameters for communication using the EN 13757-2 interfaces if the device acts as an M-bus master.

M-Bus master port setup		**0...n**	**class_id = 74, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	7	3	x + 0x08
Specific methods		***m/o***				

4.8.5.2 Attribute description

4.8.5.2.1 logical_name

Identifies the “M-Bus master port setup” object instance. See 6.2.22.

4.8.5.2.2 comm_speed

The communication speed supported by the port

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.6 DLMS®/COSEM server M-Bus port setup (class_id = 76, version = 0)

4.8.6.1 Overview

Instances of the “DLMS®/COSEM server M-Bus port setup” are used in DLMS®/COSEM servers hosted by M-Bus slave devices, using the DLMS®/COSEM wired or wireless M-Bus (wM-Bus) communication profile.

DLMS/COSEM server M-Bus port setup		0...n	class_id = 76, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. M-Bus_profile_selection	(static)	octet-string				x + 0x08
3. M-Bus_port_communication_state	(dyn.)	enum				x + 0x10
4. M-Bus_Data_Header_Type	(dyn.)	enum			2	x + 0x18
5. primary_address	(static)	unsigned			0	x + 0x20
6. identification_number	(static)	double-long-unsigned			0	x + 0x28
7. manufacturer_id	(static)	long-unsigned			0	x + 0x30
8. version	(static)	unsigned			0	x + 0x38
9. device_type	(static)	unsigned			0	x + 0x40
10. max_pdu_size	(static)	long-unsigned				x + 0x48
11. listening_window	(static)	array				x + 0x50
Specific methods		***m/o***				

4.8.6.2 Attribute description

4.8.6.2.1 logical_name

Identifies the “DLMS®/COSEM server M-Bus port setup” object instance. See 6.2.22.

4.8.6.2.2 M-Bus_profile_selection

References an M-Bus communication port setup object describing the physical capabilities for wired or wireless communication. The referenced object is either an “M-Bus slave port setup” object (class_id = 25) or a “Wireless Mode Q channel” object (class_id = 73).

4.8.6.2.3 M-Bus_port_communication_state

This attribute is relevant only in wM-Bus.

Carries the communication status of the M-Bus node.

See EN 13757-4:2013, 11.6.3, Table 27.

enum:

(0) No access: Meter provides no access windows (unidirectional meter),

(1) Temporary no access: Meter supports bidirectional access in general, but there is no access window after this transmission (e.g. temporary no access in order to keep duty cycle limits or to limit energy consumption),

(2) Limited access: Meter provides a short access windows only immediately after this transmission (e.g. battery operated meter),

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(3) Unlimited access: Meter provides unlimited access at least until next transmission (e.g. mains powered devices).

4.8.6.2.4 M-Bus_Data_ Header_Type

Carries the type of the M-Bus Data Header, derived from the CITL value from the current communication.

In the case of a push operation the default value (2) shall be applied.

enum:

(0) M-Bus_Data_Header_Type == None_M-Bus_Data_Header, the value of the CITL field shall be 0x10 when segmentation is not used, and shall be 0x00 .. 0x1F when segmentation is used (with the FIN bit set to 0 in all segments except in the last segment);

(1) M-Bus_Data_Header_Type == Short_M-Bus_Data_Header, the value of the CITL field shall be 0x61 in an M-Bus frame sent by a master and 0x7D in a an M-Bus frame sent by a slave;

(2) M-Bus_Data_Header_Type == Long_M-Bus_Data_Header, the value of the CITL field shall be 0x60 in an M-Bus frame sent by a master and 0x7C in a an M-Bus frame sent by a slave.

4.8.6.2.5 primary_address

Carries the primary address of the M-Bus slave device.

See IEC 62056-7-3:2017, Table 1.

If the slave device is already configured and thus, its primary address is different from 0, then the data exchange with the M-Bus slave device is possible.

4.8.6.2.6 identification_number

Carries the Identification Number element of the Data Header as specified in EN 13757-3:2013, 5.5.

In the case of unidirectional communication this value – together with attributes 6, 7, 8 and 9 – shall be parametrized.

In the case of bidirectional communication this attribute – together with attributes 6, 7, 8 and 9 – are filled with the values found in the first message received after installation by different M-Bus network management interfaces.

NOTE If in subsequent messages these values are not the same, the message is discarded.

4.8.6.2.7 manufacturer_id

Carries the Manufacturer Identification element of the Data Header as specified in EN 13757-3:2013, 5.6.

4.8.6.2.8 version

Carries the Version element of the Data Header as specified in EN 13757-3:2013, 5.7.

4.8.6.2.9 device_type

Carries the Device type identification element of the Data Header as specified in EN 13757-3:2013, 5.8, Table 6.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.6.2.10 max_pdu_size

Contains length capability available from M-Bus lower layers (expressed in bytes).

Specifies the maximum length of the DLMS® payload (an APDU or a part of it) that can be carried by one M-Bus frame.

In the case of long messages, either block transfer provided by the DLMS®/COSEM application layer or segmentation provided by the transport layer or both mechanisms can be used.

4.8.6.2.11 listening_window

This attribute is relevant only in wM-Bus.

Defines the time points when the point-to-point communication window(s) become active (start_time) and inactive (end_time). The start_time implicitly defines the period.

EXAMPLE When the day of month is not specified (equal to 0xFF) this means that we have a daily listening window management. Daily, monthly, etc., window management can be defined.

```
array window_element

window_element::= structure

{
        start_time: octet-string,
        end_time: octet-string
}
```

start_time and end_time are formatted as specified in 4.6.1 for *date-time*.

4.8.7 M-Bus diagnostic (class_id = 77, version = 0)

4.8.7.1 Overview

Instances of the IC “M-Bus diagnostic” hold information related to the operation of the M-Bus network, like current signal strength, channel identifier, link status to the M-Bus network and counters related to the frame exchange, transmission and frame reception quality.

M-Bus diagnostic		0...n	class_id = 77, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				X
2. received-signal-strength	(dyn.)	unsigned				x + 0x08
3. channel_Id	(dyn.)	unsigned			0	x + 0x10
4. link_status	(dyn.)	enum				x + 0x18
5. broadcast_frames_counter	(dyn.)	array			0	x + 0x20
6. transmissions_counter	(dyn.)	double-long-unsigned			0	x + 0x28
7. FCS_OK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x30
8. FCS_NOK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x38
9. capture_time	(dyn.)	structure				x + 0x40
Specific methods		***m/o***				
1. reset (data)		o				x + 0x48

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.7.2 Attribute description

4.8.7.2.1 logical_name

Identifies the “M-Bus diagnostic” object instance. See 6.2.22.

4.8.7.2.2 received_signal_strength

This attribute is relevant only for wireless bidirectional M-Bus communication.

When the wM-Bus profile is used, this attribute holds the value of signal strength of the last wM-Bus frame received, expressed in dBm.

4.8.7.2.3 channel_Id

This attribute is relevant only for wM-Bus communication.

When the wM-Bus profile is used, this attribute holds the identification of the channel currently used.

The default value is 0.

4.8.7.2.4 link_status

This attribute is relevant only for wM-Bus communication.

When the wM-Bus profile is used, this attribute holds the current status of link to the M-Bus network.

The possible stati are as specified in EN 13757-5:2015, 9.7.4.1.7 Link State.

enum:

(0) Default (data never received),
(1) Link in normal operation,
(2) Link temporarily interrupted,
(3) Link permanently interrupted

NOTE If synchronisation with network is lost, trials start automatically to re-establish the link by performing radio scans. As long as re-synchronisation attempts are in progress, the link_status is temporarily interrupted.

Link_status changes to normal operation when the link is re-established.

Link_status changes to permanently interrupted , when the manufacturer specified number of attempts is exceeded.

4.8.7.2.5 broadcast_frames_counter

Holds the broadcast-frames-counter values with time stamp of last frame received and differentiated by client identifiers.

```
array        broadcast_frame_counter_definition

broadcast_frame_counter_definition::=    structure

{
      client_id:        unsigned,
      counter:          double-long-unsigned,
      time_stamp:       date-time
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The default value is 0.

4.8.7.2.6 transmissions_counter

Counts the number of frames transmitted by the related M-Bus port.

The transmission counter is incremented at the beginning of each transmission phase. A client system can write this variable to update the counter. When the transmissions counter reaches the maximum value, it automatically returns to 0 on the next increment.

The default value is 0.

4.8.7.2.7 FCS_OK_frames_counter

Counts the number of frames received with a correct checksum. When the FCS_OK_frames_counter field reaches the maximum value, it automatically returns to 0 on the next increment.

The default value is 0.

4.8.7.2.8 FCS_NOK_frames_counter

Counts the number of frames received with an incorrect checksum. When the FCS_NOK frames counter field reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

4.8.7.2.9 capture_time

Holds the time stamp of the most recent change of the value of the attributes 2, 4, 6, 7 or 8.

```
capture_time::=   structure

{
        attribute_id:       unsigned,
        time_stamp:         date-time
}
```

4.8.7.3 Method description

4.8.7.3.1 reset (data)

Clears all counters, received_signal_strength, link_status and capture_time.

```
data::= integer(0)
```

NOTE Channel_id management is outside the scope of the specification of this IC.

See EN 13757-4:2013.

4.9 Interface classes for setting up data exchange over the Internet

4.9.1 TCP-UDP setup (class_id = 41, version = 0)

4.9.1.1 Overview

This IC allows modelling the setup of the TCP or UDP sub-layer of the COSEM TCP or UDP based transport layer of a TCP-UDP/IP based communication profile.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

In TCP-UDP/IP based communication profiles, all AAs between a physical device hosting one or more COSEM client application processes and a physical device hosting one or more COSEM server APs rely on a single TCP or UDP connection. The TCP or UDP entity is wrapped in the COSEM TCP-UDP based transport layer. Within a physical device, each AP – client AP or server logical device – is bound to a Wrapper Port (WPort). The binding is done with the help of the SAP Assignment object, see 4.4.5.

On the other hand, a COSEM TCP or UDP based transport layer may be capable to support more than one TCP or UDP connections, between a physical device and several peer physical devices hosting COSEM APs.

When a COSEM physical device supports various data link layers – for example Ethernet and PPP – an instance of the TCP-UDP setup object is necessary for each of them.

TCP-UDP setup		**0...n**	**class_id = 41, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. TCP-UDP_port	(static)	long-unsigned				x + 0x08
3. IP_reference	(static)	octet-string				x + 0x10
4. MSS	(static)	long-unsigned	40	65...535	576	x + 0x18
5. nb_of_sim_conn	(static)	unsigned	1			x + 0x20
6. inactivity_time_out	(static)	long-unsigned			180	x + 0x28
Specific methods		***m/o***				

4.9.1.2 Attribute description

4.9.1.2.1 logical_name

Identifies the “TCP-UDP setup” object instance. See 6.2.23.

4.9.1.2.2 TCP-UDP_port

Holds the TCP-UDP port number on which the physical device is listening for the DLMS®/COSEM application.

For DLMS®/COSEM, the following port numbers have been registered by the IANA. See http://www.iana.org/assignments/port-numbers.

dlms/cosem 4059/TCP DLMS/COSEM

dlms/cosem 4059/UDP DLMS/COSEM

4.9.1.2.3 IP_reference

References an IP setup object by its logical name. The referenced object contains information about the IP Address settings of the IP layer supporting the TCP-UDP layer.

4.9.1.2.4 MSS

With the help of the Maximum Segment Size (MSS) option, a TCP can indicate the maximum receive segment size to its partner. Note, that:

– this option shall only be sent in the initial connection request (i.e. in segments with the SYN control bit sent);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– if this option is not present, conventionally MSS is considered as its default value, 576;

– MSS is not negotiable; its value is indicated by this attribute.

4.9.1.2.5 nb_of_sim_conn

The maximum number of simultaneous connections the COSEM TCP-UDP based transport layer is able to support.

4.9.1.2.6 inactivity_time_out

Defines the time, expressed in seconds over which, if no frame is received from the COSEM client, the inactive TCP connection shall be aborted.

When this value is set to 0, this means that the *inactivity_time_out* is not operational. In other words, a TCP connection, once established, in normal conditions – no power failure, etc. – will never be aborted by the COSEM server.

Note, that all actions related to the management of the inactivity time-out function such as measuring the inactivity time, aborting the TCP connection if the time-out is over, etc. are managed inside the TCP-UDP layer implementation.

4.9.2 IPv4 setup (class_id = 42, version = 0)

4.9.2.1 Overview

NOTE Compared to earlier editions of this document, this specification provides improvements in presenting the attributes. As this does not constitute technical changes, the version of the IC remains 0.

This IC allows modelling the setup of the IPv4 layer, handling all information related to the IP Address settings associated to a given device and to a lower layer connection on which these settings are used.

There shall be an instance of this IC in a device for each different network interface implemented. For example, if a device has two interfaces (using the TCP-UDP/IPv4 profile on both of them), there shall be two instances of the IPv4 setup IC in that device: one for each of these interfaces.

IPv4 setup	**0...n**	**class_id = 42, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. DL_reference (static)	octet-string				x + 0x08
3. IP_address	double-long-unsigned				x + 0x10
4. multicast_IP_address	array				x + 0x18
5. IP_options	array				x + 0x20
6. subnet_mask	double-long-unsigned				x + 0x28
7. gateway_IP_address	double-long-unsigned				x + 0x30
8. use_DHCP_flag (static)	boolean				x + 0x38
9. primary_DNS_address	double-long-unsigned				x + 0x40
10. secondary_DNS_address	double-long-unsigned				x + 0x48
Specific methods	***m/o***				
1. add_mc_IP_address (data)	o				x + 0x60
2. delete_mc_IP_address (data)	o				x + 0x68
3. get_nbof_mc_IP_addresses (data)	o				x + 0x70

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.2.2 Attribute description

4.9.2.2.1 logical_name

Identifies the “IPv4 setup” object instance. See 6.2.23.

4.9.2.2.2 DL_reference

References a Data link layer (e.g. Ethernet or PPP) setup object by its logical name.

The referenced object contains information about the specific settings of the data link layer supporting the IP layer.

4.9.2.2.3 IP_address

Carries the value of the IP address (IPv4) of this physical device on the network to which the device is connected via the referenced lower layer interface. It can be either (static) or (dynamic). In the latter case, dynamic IP address assignment (for example DHCP) is used.

If no IP address is assigned, the value is 0.

EXAMPLE The IPv4 address 192.168.0.1 (in dotted decimal notation) corresponds to C0A80001 (hexa) which gives 3232235521 (double-long-unsigned).

4.9.2.2.4 multicast_IP_address

Contains an array of IP addresses. IP addresses in this array shall fall into the multicast group address range (“Class D” addresses, including IP addresses in the range of 224.0.0.0 to 239.255.255.255). When a device receives an IP datagram with one of these IP addresses in the destination IP address field, it shall consider that this datagram is addressed to it.

```
multicast_IP_address::= array double-long-unsigned
```

4.9.2.2.5 IP_options

Contains the necessary parameters to support the selected IP options – for example Datagram time-stamping or security services (IPSec).

```
IP_options::= array      IP_options_element

IP_options_element::= structure
{
        IP_Option_Type:         unsigned,
        IP_Option_Length:       unsigned,
        IP_Option_Data:         octet-string
}
```

NOTE In all cases, as specified in RFC 791, the IP_Option_Length field includes the total length of all three fields: IP_Option_Type, IP_Option_Length and IP_Option_Data.

Allowed IP_Option_Types:

a) Security: IP_Option_Type = 0x82, IP_Option_Length = 11

If this option is present, the device shall be allowed to send security, compartmentation, handling restrictions and TCC (closed user group) parameters within its IP Datagrams. The IP_Option_Data shall contain the value of the Security, Compartments, Handling Restrictions and Transmission Control Code values, as specified in RFC 791.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

b) Loose Source and Record Route: IP_Option_Type = 0x83

If this option is present, the device shall supply routing information to be used by the gateways in forwarding the datagram to the destination, and to record the route information. The IP_Option_Length and IP_Option_Data values are specified in RFC 791.

c) Strict Source and Record Route: IP_Option_Type = 0x89

If this option is present, the device shall supply routing information to be used by the gateways in forwarding the datagram to the destination, and to record the route information. The IP_Option_Length and IP_Option_Data values are specified in RFC 791.

d) Record Route: IP_Option_Type = 0x07

If this option is present, the device shall additionally:

– send originated IP Datagrams with that option, providing means to record the route of these Datagrams;
– as a router, send routed IP Datagrams with the route option adjusted according to this option.

The IP_Option_Length and IP_Option_Data values are specified in RFC 791.

e) Internet Timestamp: IP_Option_Type = 0x44

If this option is present, the device shall additionally:

– send originated IP Datagrams with that option, providing means to time-stamp the datagram in the route to its destination;
– as a router, send routed IP Datagrams with the time-stamp option adjusted according to this option.

The IP_Option_Length and IP_Option_Data values are specified in RFC 791.

4.9.2.2.6 subnet_mask

Contains the subnet mask.

When sub-networking is used in a network segment, each device concerned shall behave conforming to the sub-networking rules. In order to do that, the device, besides of its IP address, needs also to know, how the IP address is structured within this sub-networked segment. The subnet_mask attribute carries this information.

With IPv4, the *subnet_mask* is a 32 bits word, expressed exactly in the same format as an IP Address (for example 255.255.255.0), but has another meaning: the '0' bits of the *subnet_mask* indicate the portion of the IP Address which is still used as Device_ID on a sub-networked IP Network[3].

4.9.2.2.7 gateway_IP_address

Contains the IP Address of the gateway device.

In most IP implementations, there is code in the module that handles outgoing datagrams to decide if a datagram can be sent directly to the destination on the local network or if it shall be sent to a gateway. In order to be able to send non-local datagrams to the gateway, the device shall know the IP address of the gateway device assigned to the given network segment.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If no IP address is assigned, the value is 0.

4.9.2.2.8 use_DHCP_flag

TRUE: The device uses DHCP (Dynamic Host Configuration Protocol) to dynamically determine the IP_address, subnet_mask and *gateway_IP_address* parameters.

FALSE: The *IP_address, subnet_mask* and *gateway_IP_address* parameters shall be set locally.

4.9.2.2.9 primary_DNS_address

The IP Address of the primary Domain Name Server (DNS).

If no IP address is assigned, the value is 0.

4.9.2.2.10 secondary_DNS_ address

The IP Address of the secondary Domain Name Server (DNS).

If no IP address is assigned, the value is 0.

4.9.2.3 Method description

4.9.2.3.1 add_mc_IP_address (IP_Address)

Adds one multicast IP address to the *multicast_IP_address* array.

IP_Address::= double-long-unsigned

4.9.2.3.2 delete_mc_IP_address (IP_Address)

Deletes one IP Address from the *multicast_IP_address* array. The IP Address to be deleted is identified by its value.

IP_Address::= double-long-unsigned

4.9.2.3.3 get_nbof_mc_IP_addresses (data)

Returns the number of IP Addresses contained in the *multicast_IP_address* array.

data::= unsigned

4.9.3 IPv6 setup (class_id = 48, version = 0)

4.9.3.1 Overview

NOTE See also Annex C.

The IPv6 setup IC allows modelling the setup of the IPv6 layer, handling all information related to the IPv6 address settings associated to a given device and to a lower layer connection on which these settings are used.

There shall be an instance of this IC in a device for each different network interface implemented. For example, if a device has two interfaces (using the UDP/IP and/or TCP/IP profile on both of them), there shall be two instances of the IPv6 setup IC in that device: one for each of these interfaces.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IPv6 setup		0...n	class_id = 48, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. DL_reference	(static)	octet-string				x + 0x08
3. address_config_mode	(static)	enum			0	x + 0x10
4. unicast_IPv6_addresses		array				x + 0x18
5. multicast_IPv6_addresses	(static)	array				x + 0x20
6. gateway_IPv6_addresses	(static)	array			0	x + 0x28
7. primary_DNS_address	(static)	octet-string			0	x + 0x30
8. secondary_DNS_address	(static)	octet-string			0	x + 0x38
9. traffic_class	(static)	unsigned	0	63	0	x + 0x40
10. neighbor_discovery_setup	(static)	array				x + 0x48
Specific methods		***m/o***				
1. add_IPv6_address (data)		*o*				x + 0x60
2. remove_IPv6_address (data)		*o*				x + 0x68

4.9.3.2 Attribute description

4.9.3.2.1 logical_name

Identifies the “IPv6 setup” object instance. See 6.2.23.

4.9.3.2.2 DL_reference

References a Data link layer setup object by its logical name. The referenced object contains information about the specific settings of the data link layer supporting the IPv6 layer.

4.9.3.2.3 address_config_mode

Defines the IPv6 address configuration mode

enum:

(0) Auto-configuration (default),
(1) DHCPv6,
(2) Manual,
(3) ND (Neighbour Discovery)

NOTE The address_config_mode is common for all IPv6 addresses managed by an instance of the IPv6 setup class.

4.9.3.2.4 unicast_IPv6_addresses

Carries unicast IPv6 address(es) assigned to the related interface of the physical device on the network (unique local unicast, link local unicast and / or global unicast addresses). An IPv6 address can be either (static) or (dynamic) or both.

unicast_IPv6_addresses::= array octet-string

The format of each unicast IPv6 address shall be as specified in RFC 3513.

If no unicast IPv6 address is assigned to the interface, an array of zero elements is present (default).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

To reset all unicast IPv6 address(es) configured, an array of zero elements shall be written.

4.9.3.2.5 multicast_IPv6_addresses

Contains an array of IPv6 addresses used for multicast.

multicast_IPv6_addresses::= array octet-string

The format of each multicast IPv6 address shall be as specified in RFC 3513 .

If no multicast IPv6 address is assigned to the interface, an array of zero elements is present (default).

To reset all multicast IPv6 address(es) configured, an array of zero elements shall be written.

4.9.3.2.6 gateway_IPv6_addresses

Contains the IPv6 addresses of the IPv6 gateway device.

gateway_IPv6_addresses::= array octet-string

The format of each gateway IPv6 address shall be as specified in RFC 3513.

If no gateway IPv6 address is assigned to the interface, an array of zero elements is present (default).

To reset all gateway IPv6 address(es) configured, an array of zero elements shall be written.

4.9.3.2.7 primary_DNS_address

Contains the IPv6 address of the primary Domain Name Server (DNS).

If no IPv6 address is assigned, the length of the octet-string shall be 0.

4.9.3.2.8 secondary_DNS_address

Contains the IPv6 address of the secondary Domain Name Server (DNS).

If no IPv6 address is assigned, the length of the octet-string shall be 0.

4.9.3.2.9 traffic_class

Contains the traffic class element of the IPv6 header. The 6 most-significant bits are used for DSCP (Differentiated Services Codepoint), which is used to classify packets as specified in RFC 2474:1998, Clause 3.

4.9.3.2.10 neighbor_discovery_setup

Contains the configuration to be used for both routers and hosts to support the Neighbor Discovery protocol for IPv6 (RFC 4861).

array neighbor_discovery_setup

neighbor_discovery_setup::= structure

{
RS_max_retry: unsigned,
RS_retry_wait_time: long-unsigned,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
        RA_send_period:     double-long-unsigned
    }
```

Where:

RS_max_retry — Gives the maximum number of router solicitation retries to be performed by a node if the expected router advertisement has not been received.
Range: 1-255
Default value: 3

RS_retry_wait_time — Gives the waiting time in milliseconds between two successive router solicitation retries.
Range: 0-65 535
Default value: 10 000

RA_send_period — Gives the router advertisement transmission period in seconds.
Range: 0-4 294 967 295

4.9.3.3 Method description

4.9.3.3.1 add_IPv6_address (data)

Adds one IPv6 address for the physical interface to the IPv6 address array.

```
data::= structure
{
    IPv6_address_type:    enum,
    IPv6_address:         octet-string
}
```

Where IPv6_address_type defines the type of IPv6 address to add:

enum:

(0) unicast,
(1) multicast,
(2) gateway

IPv6_address specifies the IPv6 address to add. The new address will be automatically added at the end of the list. If an address has to be added to a specific position, this can be done by writing the whole array.

4.9.3.3.2 remove_IPv6_address (data)

Removes one IPv6 address for the physical interface to the IPv6 address array.

```
data::= structure

{
IPv6_address_type:    enum,
IPv6_address:         octet-string
}
```

Where IPv6_address_type defines the type of IPv6 address to remove:

enum:

(0) unicast,
(1) multicast,
(2) gateway

IPv6_address specifies the IPv6 address to remove.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.4 MAC address setup (class_id = 43, version = 0)

4.9.4.1 Overview

NOTE 1 The name and the use of this interface class has been changed from “Ethernet setup” to “MAC address setup” to allow a more general use, without changing the version.

Instances of this IC hold the MAC address of the physical device (or, more generally, a device or software.) There shall be an instance of this IC for each network interface of a physical device.

NOTE 2 In the case of the three-layer HDLC based communication profile, the MAC address (lower HDLC address) is carried by an IEC HDLC setup object.

NOTE 3 In the case of the S-FSK PLC communication profile, the MAC address is carried by a S-FSK Phy&MAC setup object.

MAC address setup	0...n	class_id = 43, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. MAC_address	octet-string				x + 0x08
Specific methods	***m/o***				

4.9.4.2 Attribute description

4.9.4.2.1 logical_name

Identifies the “MAC address setup” object instance. See 6.2.21, 6.2.23 and 6.2.27.

4.9.4.2.2 mac_address

Holds the MAC address.

4.9.5 PPP setup (class_id = 44, version = 0)

4.9.5.1 Overview

NOTE Compared to earlier editions of this document, this specification provides improvements in presenting the attributes. As this does not constitute technical changes, the version of the IC remains 0.

This IC allows modelling the setup of interfaces using the PPP protocol, by handling all information related to PPP settings associated to a given physical device and to a lower layer connection on which these settings are used. There shall be an instance of this IC for each network interface of a physical device.

PPP setup	0...n	class_id = 44, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. PHY_reference (static)	octet-string				x + 0x08
3. LCP_options (static)	LCP_options_type				x + 0x10
4. IPCP_options (static)	IPCP_options_type				x + 0x18
5. PPP_authentication (static)	PPP_auth_type				x + 0x20
Specific methods	***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.5.2 Attribute description

4.9.5.2.1 logical_name

Identifies the “PPP setup” object instance. See 6.2.23.

4.9.5.2.2 PHY_reference

References another object by its logical_name. The object referenced contains information about the specific physical layer interface, supporting the PPP layer.

4.9.5.2.3 LCP_options

Contains the necessary parameters to support the selected LCP configuration options.

```
LCP_options_type::= array LCP_options_type_element

LCP_options_type_element::=     structure

{
      LCP_Option_Type:          unsigned,
      LCP_Option_Length:        unsigned,
      LCP_Option_Data:          CHOICE
      {
            structure           [2] -- for Callback-data
            boolean             [3] -- for ProtF-Compr and AdCtr-Compr,
            double-long-unsigned [6] -- for ACCM and Mag-Num,
            unsigned            [17] -- for FCS-Alternatives,
            long-unsigned       [18] -- for MRU and Auth-Prot
      }
}
```

NOTE 1 In all cases, as specified in IETF STD 51 / RFC 1661, the LCP_Option_Length field includes the total length of all three fields: LCP_Option_Type, LCP_Option_Length and LCP_Option_Data.

NOTE 2 For assigned values see Point-to-Point (PPP) Protocol Field Assignments, available at: http://www.iana.org/assignments/ppp-numbers/ppp-numbers.xml

The supported LCP_Option_Types are the following:

a) Maximum-Receive-Unit (MRU), LCP_Option_Type = 1. See IETF STD 51 / RFC 1661.

This configuration option may be sent to inform the peer that the implementation can receive larger packets, or to request that peer send smaller packets. The default value is 1 500 octets;

b) Async-Control-Character-Map (ACCM), LCP_Option_Type = 2. See IETF STD 51 / RFC 1662.

This configuration option provides a method to negotiate the use of control character transparency on asynchronous links;

c) Authentication-Protocol, LCP_Option_Type = 3. See IETF STD 51 / RFC 1661.

This configuration option provides a method to negotiate the use of a specific protocol for authentication. By default, authentication is not required. The value indicates the authentication protocol used on the given PPP link. Possible values are:

0x0000 – No authentication protocol is used,

0xc023 – The PAP protocol is used,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

0xc223 – The CHAP protocol is used,

0xc227 – The EAP protocol is used.

d) Magic-Number, LCP_Option_Type = 5. See IETF STD 51 / RFC 1661.

This configuration option provides a method to detect looped-back links and other data link layer anomalies;

e) Protocol-Field-Compression (PFC), LCP_Option_Type = 7. See IETF STD 51 / RFC 1661.

This configuration option provides a method to negotiate the compression of the PPP protocol fields;

f) Address-and-Control-Field-Compression (ACFC), LCP_Option_Type = 8. See IETF STD 51 / RFC 1661.

This configuration option provides a method to negotiate the compression of the data link layer address and control fields;

g) FCS-Alternatives, LCP_Option_Type = 9. See RFC 1570.

This configuration option provides a method for an implementation to specify another FCS format to be sent by the peer, or to negotiate away the FCS altogether. The value of the FCS-Alter (FCS Alternatives) options field identifies the FCS used. This field is one octet, and is comprised of the "logical or" of the following values:

bit 1 Null FCS,

bit 2 CCITT 16-bit FCS,

bit 4 CCITT 32-bit FCS

h) Callback, LCP_Option_Type = 13. See RFC 1570.

This configuration option provides a method for an implementation to request a dial-up peer to call back. This provides enhanced security by ensuring that the remote site can connect only from a single location as defined by the callback number.

```
callback_data::= structure

{
        callback_active:            boolean, // default: false,
        callback_data_length:       unsigned,
        callback_operation:         unsigned,
        callback_message:           octet-string
}
```

Where:

– the callback-active field indicates whether the callback option is active on this PPP link;

– the callback_operation field indicates the contents of the Message field;

callback_operation: unsigned

(0) Location determined by user authentication,
(1) Dialling string,
(2) Location identifier,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(3) E.164 number,
(4) X.500 distinguished name,
(5) Unassigned,
(6) Location is determined during CBCP negotiation.

the callback_message field is zero or more octets, and its general contents are determined by the callback_operation field. The actual format of the information is site or application specific.

4.9.5.2.4 IPCP_options

Contains the necessary parameters for the IP Control Protocol – the Network Control Protocol module of the PPP – that allow the negotiation of desirable Internet Protocol parameters. For details on IPCP, please refer to RFC 1332.

```
IPCP_options_type::= array   IPCP_options_type_element

IPCP_options_type_element::=     structure

{
      IPCP_Option_Type:          unsigned,
      IPCP_Option_Length:        unsigned,
      IPCP_Option_Data:          CHOICE
      {
            array                                  [1] -- for Pref-Peer-IP,
            -- each IP address is of type double-long-unsigned
            boolean                                [3] -- for GAO and USIP,
            double-long-unsigned                   [6] -- for Pref-Local-IP,
            long-unsigned                          [18] -- for IP-Comp-Prot
      }
}
```

NOTE 1 In all cases, as specified in RFC 1332, the IPCP_Option_Length field includes the total length of all three fields: IPCP_Option_Type, IPCP_Option_Length and IPCP_Option_Data.

The supported IPCP_Option_Types are the following:

a) IP-Compression-Protocol (IP-Comp-Prot) IPCP_Option_Type = 2. See RFC 1332.

This configuration option provides a way to negotiate the use of a specific compression protocol. By default, compression is not enabled. Possible values are:

0x0000 – No IP Compression is used (default),
0x002d – Van Jacobson Compressed TCP/IP (RFC 1332),
0x0003 – Robust Header Compression (ROHC) (RFC 3241),
0x0061 – IP Header Compression (RFC 2507, RFC 3544)

b) Preferred-Local-IP-Address (Pref-Local-I), IPCP_Option_Type = 3. See RFC 1332.

This configuration option provides a way to negotiate the IP address to be used on the local end of the link. It allows the sender of the Configure-Request to state, which IP-address is desired, or to request that the peer provide the information. The peer can provide this information by NAK-ing the option, and returning a valid IP-Address;

NOTE 2 In RFC 1332 the name of option Type 3 is “IP-Address”.

NOTE 3 Option types 20, 21 and 22 have been specified for the purposes of DLMS®/COSEM; they are not specified in RFC 1332.

c) Preferred-Peer-IP-Addresses (Pref-Peer-IP), IPCP_Option_Type = 20.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

This configuration option provides a way to negotiate the IP Address to be used on the remote end of the link. When the Grant-Access-Only-to-Pref-Peer-on-List (GAO) parameter is set to be TRUE, the device shall accept PPP connection only with a remote device having one of the IP Addresses on this list. When the Use-Static-IP-Pool (USIP) parameter is set to TRUE, the COSEM Server device shall try to assign one of these IP Addresses to the remote device;

d) Grant-Access-Only-to-Pref-Peer-on-List (GAO), IPCP_Option_Type = 21.

This configuration option indicates whether the device can accept PPP connection only with peer devices with IP Address on the above list or not. Its default value is FALSE;

e) Use-Static-IP-Pool (USIP), IPCP_Option_Type = 22.

This configuration option indicates whether the device should try to assign one of the IP Addresses of the Preferred-Peer-IP-Addresses to the remote end device during the IP Address negotiation phase or not.

4.9.5.2.5 PPP_authentication

Contains the parameters required by the PPP authentication procedure used.

```
PPP_auth_type::=        CHOICE

{
        null-data       [0] -- used when no authentication is required,
        structure       [2] -- PAP_login or CHAP_algorithm or EAP_params
}

PAP_login::=        structure
{
        user-name:          octet-string,
        PAP-password:       octet-string
}


CHAP_algorithm::=       structure
{
        user-name:          octet-string,
        algorithm_id:       unsigned
}
```

Possible values for CHAP-algorithm-id parameter today are as follows:
0x05 – CHAP with MD5 (default), see RFC 1994.
0x06 – SHA-1,
0x80 – MS-CHAP, see RFC 2433
0x81 – MS-CHAP-2, see RFC 2759.

NOTE 1 New values can be used as become assigned.

When CHAP is used, a “secret” is also required to verify the “challenge” sent by the client. This “secret” is not accessible in the PPP setup object.

NOTE 2 The PPP Challenge Handshake Authentication Protocol (CHAP) is specified in RFC 1994.

```
EAP_params::= structure

{
        md5_challenge (Type 4):                 boolean,
        one_time_password (OTP, Type 5):        boolean,
        generic_token_card (GTC, Type 6):       boolean
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE 3 The Extensible Authentication Protocol (EAP) is specified in RFC 3748.

4.9.6 SMTP setup (class_id = 46, version = 0)

4.9.6.1 Overview

This IC allows setting up data exchange using the SMTP protocol.

SMTP modem setup		**0...n**	**class_id = 46, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. server_port	(static)	long-unsigned			25	x + 0x08
3. user_name	(static)	octet-string				x + 0x10
4. login_password	(static)	octet-string				x + 0x18
5. server_address	(static)	octet-string				x + 0x20
6. sender_address	(static)	octet-string				x + 0x28
Specific methods		***m/o***				

4.9.6.2 Attribute description

4.9.6.2.1 logical_name

Identifies the “SMTP setup” object instance. See 6.2.23.

4.9.6.2.2 server_port

Defines the value of the TCP-UDP port related to this protocol. By default, this value is the SMTP port numberID assigned by IANA:

smtp 25/tcp, smtp 25/udp Simple Mail Transfer

4.9.6.2.3 user_name

Defines the user name to be used for the login to the SMTP server.

4.9.6.2.4 login_password

Password to be used for login. When the string is void, this means that there is no authentication.

4.9.6.2.5 server_address

Defines the server address as an octet string. This server address can be a name, which shall be resolvable by the primary DNS or the secondary DNS. In the case when it is directly the IP address of the server, which is specified here, it shall be a string in dotted format. EXAMPLE: 163.187.45.87.

4.9.6.2.6 sender_address

Defines the sender address as an octet string. This sender address can be a name. In the case when it is directly the IP address of the sender, which is specified here, it will be a string in dotted format.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.7 NTP setup (class_id = 100, version = 0)

4.9.7.1 Overview

Instances of the “NTP setup” IC allow setting up time synchronisation using the NTP protocol as specified in RFC 5905. One or several instances may be configured to support multiple time servers.

NTP Setup		0...n	class_id = 100, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. activated	(static)	boolean			FALSE	x + 0x08
3. server_address	(static)	octet-string				x + 0x10
4. server_port	(static)	long-unsigned			123	x + 0x18
5. authentication_method	(static)	enum				x + 0x20
6. authentication_keys	(static)	array				x + 0x28
7. client_key	(static)	octet-string				x + 0x30
Specific methods		***m/o***				
1. synchronize (data)		m				x + 0x38
2. add_authentication_key (data)		o				x + 0x40
3. delete_authentication_key (data)		o				x + 0x48

4.9.7.2 Attribute description

4.9.7.2.1 General

logical_name Identifies the “NTP setup” object instance. See 6.2.23.

4.9.7.2.2 activated

Defines if the NTP time synchronisation is active or not.

Synchronisation active = TRUE

4.9.7.2.3 server_address

Defines the NTP server address as an octet string. This server address can be a name, which shall be resolvable by the primary DNS or the secondary DNS. In the case when it is directly the IP address of the server the dot-decimal notation shall be used for IPv4 addresses and the text format for IPv6 addresses.

Examples: 163.187.45.87 (IPv4) or 2001:db8::1:0:0:1 (IPv6).

4.9.7.2.4 server_port

Defines the value of the UDP port related to this protocol. By default, this value is the NTP port number ID assigned by IANA:

ntp 123/udp Network Time Protocol

4.9.7.2.5 authentication_ method

Defines the authentication mode used for NTP protocol.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

enum:

(0) no_security,
(1) shared_secrets,
(2) auto_key_IFF

4.9.7.2.6 authentication_keys

Contains the necessary symmetric keys if shared secrets mode of authentication is used.

authentication_keys::= array authentication_key

authentication_key::= structure

{
key_id double-long-unsigned,
key octet-string
}

4.9.7.2.7 client_key

Specifies the client key (NTP server public key) for NTP auto key authentication mechanism using IFF authentication scheme (auto_key_IFF).

4.9.7.3 Method description

4.9.7.3.1 synchronize (data)

Synchronizes the time of the DLMS® server with the NTP server.

data::= integer(0)

4.9.7.3.2 add_authentication_key (data)

Adds a new symmetric authentication key to authentication key array.

data::= structure

{
key_id double-long-unsigned,
key octet-string
}

4.9.7.3.3 delete_authentication_key (data)

Deletes a symmetric authentication key from the key array. The key to be deleted is identified by its key_id.

data::= double-long-unsigned

4.10 Interface classes for setting up data exchange using S-FSK PLC

4.10.1 General

This subclause specifies COSEM interface classes to set up and manage the protocol layers of DLMS®/COSEM S-FSK PLC communication profile:

- the S-FSK Physical layer and the MAC sub-layer as defined in IEC 61334-5-1:2001 and IEC 61334-4-512:2001;
- the LLC sub-layer as specified in IEC 61334-4-32:1996.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The MIB variables / logical link parameters specified in IEC 61334-4-512:2001, IEC 61334-5-1:2001, IEC 61334-4-32:1996 and ISO/IEC 8802-2:1998 respectively have been mapped to attributes and/or methods of COSEM ICs. The specification of these elements has been taken from the above standards and the text has been adapted to the DLMS®/COSEM environment.

NOTE IEC 61334-4-512:2001 also specifies some management variables to be used on the Client side. However, the Client side object model is not covered in this document.

For definitions related to S-FSK PLC profile see 3.2.

4.10.2 Overview

COSEM objects for setting up the S-FSK PLC channel and the LLC layer, if implemented, shall be located in the Management Logical Device of COSEM servers.

Figure 27 shows an example with a COSEM physical device comprising three logical devices. Each logical device shall contain a Logical Device Name (LDN) object. Each logical device contains one or more Association objects, one for each client supported.

NOTE As in this example there is more than one logical device, the mandatory Management Logical Device contains a SAP Assignment object instead of a Logical Device Name object.

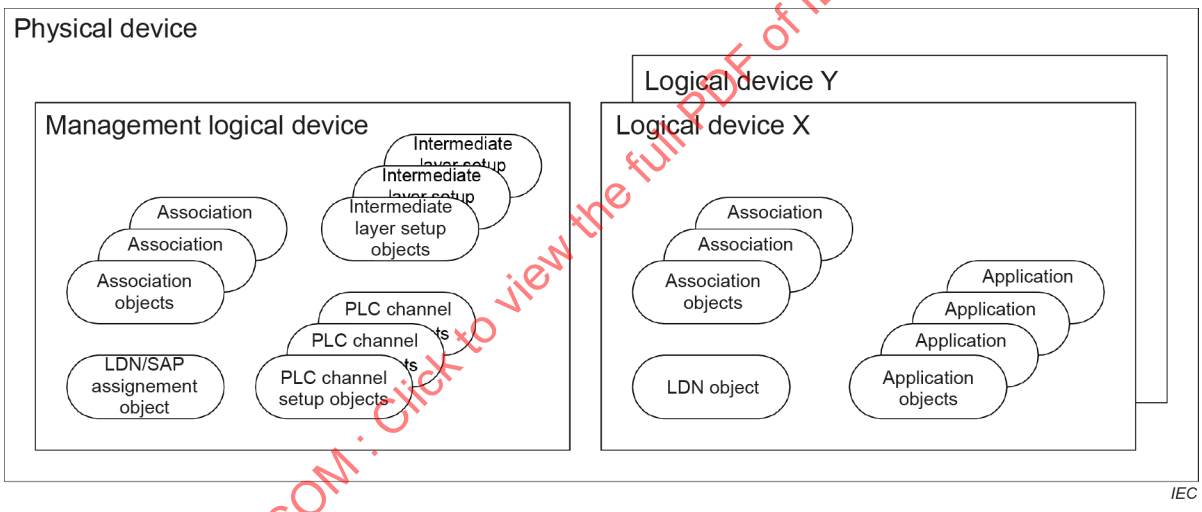


Figure 27 – Object model of DLMS®/COSEM servers

The management logical device contains the setup objects of the physical and MAC layers of the PLC channel, as well as setup objects for the intermediate layer(s). It may contain further application objects.

The other logical devices, in addition to the Association and Logical Device Name objects mentioned above, contain further application objects, holding parameters and measurement values.

IEC 61334-4-512:2001 uses DLMS® named variables to model the MIB objects and specifies their DLMS® name in the range 8…184. For compatibility with existing implementations, the short names 8…400 [sic] are reserved for devices using the IEC 61334-5-1:2001 S-FSK PLC profiles without COSEM. Therefore, when mapping the attributes and methods of the COSEM objects specified in this document to DLMS® named variables (SN mapping) this range shall not be used.

Table 38 shows the mapping of MIB variables to attributes and/or methods of COSEM ICs.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Note that on the one hand, not all MIB variables specified in IEC 61334-4-512:2001 have been mapped to attributes and methods of COSEM ICs. On the other hand, some new management variables are specified in this document.

Table 38 – Mapping IEC 61334-4-512:2001 MIB variables to COSEM IC attributes / methods

Name	Reference (unless otherwise indicated)	Interface class	class_id / attribute / method
S-FSK Physical layer management			
delta-electrical-phase	variable 1	S-FSK Phy&MAC set-up (class_id = 50, version = 1)	50 / Attr. 3
max-receiving-gain	variable 2		50 / Attr. 4
max-transmitting-gain	–		50 / Attr. 5
search-initiator-threshold	–		50 / Attr. 6
frequencies	–		50 / Attr. 7
transmission-speed	–		50 / Attr. 15
MAC layer management			
mac-address	variable 3	S-FSK Phy&MAC set-up (class_id = 50, version = 1)	50 / Attr. 8
mac-group-addresses	variable 4		50 / Attr. 9
repeater	variable 5		50 / Attr. 10
repeater-status	–		50 / Attr. 11
search-initiator time-out	–	S-FSK MAC synchronization timeouts (class_id = 52, version = 0)	52 / Attr. 2
synchronization-confirmation-time-out	variable 6		52 / Attr. 3
time-out-not-addressed	variable 7		52 / Attr. 4
time-out-frame-not-OK	variable 8		52 / Attr. 5
min-delta-credit	variable 9	S-FSK Phy&MAC set-up (class_id = 50, version = 1)	50 / Attr. 12
initiator-mac-address	IEC 61334-5-1:2001 4.3.7.6		50 / Attr. 13
synchronization-locked	variable 10		50 / Attr. 14
IEC 61334-4-32 LLC layer management			
max-frame-length	IEC 61334-4-32:1996 5.1.4	IEC 61334-4-32 LLC setup (class_id = 55, version = 1)	55 / Attr. 2
reply-status-list	variable 11		55 / Attr. 3
broadcast-list	variable 12	–	–
L-SAP-list	variable 13	NOTE In DLMS®/COSEM, L-SAPs of logical devices are held by a SAP Assignment object	
ACSE management			
application-context-list	variable 14	NOTE In DLMS®/COSEM the Association objects play a similar role.	
Application management			
active-initiator	variable 15	S-FSK Active initiator (class_id = 51, version = 0)	51 / Attr. 2
MIB system objects			
reporting-system-list	variable 16	S-FSK Reporting system list (class_id = 56, version = 0)	56 / Attr. 2

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Name	Reference (unless otherwise indicated)	Interface class	class_id / attribute / method
Other MIB objects			
reset-NEW-not-synchronized	variable 17	S-FSK Active initiator (class_id = 51, version = 0)	51 / Method 1
new-synchronization	IEC 61334-5-1:2001 4.3.7.6	–	
initiator-electrical-phase	variable 18		50 / Attr. 2
broadcast-frames-counter	variable 19	S-FSK MAC counters (class_id = 53, version = 0)	53 / Attr. 4
repetitions-counter	variable 20		53 / Attr. 5
transmissions-counter	variable 21		53 / Attr. 6
CRC-OK-frames-counter	variable 22		53 / Attr. 7
CRC-NOK-frames-counter	–		53 / Attr. 8
synchronization-register	variable 23		53 / Attr. 2
desynchronization-listing	variable 24		53 / Attr. 3

4.10.3 S-FSK Phy&MAC set-up (class_id = 50, version = 1)

4.10.3.1 Overview

NOTE The use of version 0 of this interface class is deprecated.

An instance of the “S-FSK Phy&MAC set-up” class stores the data necessary to set up and manage the physical and the MAC layer of the PLC S-FSK lower layer profile.

S-FSK Phy&MAC setup		**0...n**	**class_id = 50, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. initiator_electrical_phase	(static)	enum	0	3		x + 0x08
3. delta_electrical_phase	(dyn.)	enum	0	6		x + 0x10
4. max_receiving_gain	(static)	unsigned				x + 0x18
5. max_transmitting_gain	(static)	unsigned				x + 0x20
6. search_initiator_threshold	(static)	unsigned			98	x + 0x28
7. frequencies	(static)	frequencies_type				x + 0x30
8. mac_address	(dyn.)	long-unsigned			FFE	x + 0x38
9. mac_group_addresses	(static)	array				x + 0x40
10. repeater	(static)	enum				x + 0x48
11. repeater_status	(dyn.)	boolean				x + 0x50
12. min_delta_credit	(dyn.)	unsigned				x + 0x58
13. initiator_mac_address	(dyn.)	long-unsigned				x + 0x60
14. synchronization_locked	(dyn.)	boolean				x + 0x68
15. transmission_speed	(static)	enum	0	6	3	x + 0x70
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2 Attribute description

4.10.3.2.1 logical_name

Identifies the “S-FSK Phy&MAC setup” object instance. See 6.2.24.

4.10.3.2.2 initiator_electrical_phase

Holds the MIB variable initiator-electrical-phase (variable 18) specified in IEC 61334-4-512:2001, 5.8.

It is written by the client system to indicate the phase to which it is connected.

enum:

(0) Not defined (default),
(1) Phase 1,
(2) Phase 2,
(3) Phase 3

NOTE This enumeration is different from that of IEC 61334-4-512.

4.10.3.2.3 delta_electrical_phase

Holds the MIB variable *delta-electrical-phase* (variable 1) specified in IEC 61334-4-512:2001, 5.2 and IEC 61334-5-1:2001, 3.5.5.3.

It indicates the phase difference between the client's connecting phase and the server's connecting phase. The following values are predefined:

enum:

(0) Not defined: the server is temporarily not able to determine the phase difference,
(1) The server system is connected to the same phase as the client system.

The phase difference between the server's connecting phase and the client's connecting phase is equal to:

(2) 60 degrees,
(3) 120 degrees,
(4) 180 degrees,
(5) -120 degrees,
(6) -60 degrees

4.10.3.2.4 max_receiving_gain

Holds the MIB variable *max-receiving-gain* (variable 2) specified in IEC 61334-4-512:2001, 5.2 and IEC 61334-5-1:2001, 3.5.5.3.

Corresponds to the maximum allowed gain bound to be used by the server system in the receiving mode. The default unit is dB.

NOTE In IEC 61334-4-512:2001, no units are specified.

The possible values of the gain may depend on the hardware. Therefore, after writing a value to this attribute, the value should be read back to know the actual value.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2.5 max_transmitting_gain

Holds the value of the *max-transmitting-gain*.

Corresponds to the maximum attenuation bound to be used by the server system in the transmitting mode. The default unit is dB.

The possible values of the gain may depend on the hardware. Therefore, after writing a value to this attribute, the value should be read back to know the actual value.

4.10.3.2.6 search_initiator_threshold

This attribute is used in the intelligent search initiator process. If the value of the initiator signal is above the value of this attribute, a fast synchronization process is possible.

The default value is 98 dBµV.

4.10.3.2.7 frequencies

Contains frequencies required for S-FSK modulation.

```
frequencies_type::= structure

{
        mark_frequency:         double-long-unsigned,
        space_frequency:        double-long-unsigned
}
```

The default unit is Hz.

4.10.3.2.8 mac_address

Holds the MIB variable *mac-address* (variable 3) specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6..

NOTE 1 MAC addresses are expressed on 12 bits.

Contains the value of the address of the physical attachment (MAC address) associated to the local system. In the unconfigured state, the MAC address is “NEW-address”.

This attribute is locally written by the CIASE when the system is registered (with a Register service). The value is used in each outgoing or incoming frame. The default value is "NEW-address".

This attribute is set to NEW:

– by the MAC sub-layer, once the time-out-not-addressed delay is exceeded;
– when a client system “resets” the server system. See 4.10.4.

When this attribute is set to NEW:

– the system loses its synchronization (function of the MAC-sublayer);
– the *mac_group_address* attribute is reset (array of 0 elements);
– the system automatically releases all AAs which can be released.

NOTE 2 The second item is not present in IEC 61334-4-512:2001.

The predefined MAC addresses are shown in Table 39.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2.9 mac_group_addresses

Holds the MIB variable *mac-group-address* (variable 4) specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

Contains a set of MAC group addresses used for broadcast purposes.

array mac-address
mac-address::= long-unsigned

The ALL-configured-address, ALL-physical-address and NO-BODY addresses are not included in this list. These ones are internal predefined values.

This attribute shall be written by the initiator using DLMS® services to declare specific MAC group addresses on a server system.

This attribute is locally read by the MAC sublayer when checking the destination address field of a MAC frame not recognized as an individual address or as one of the three predefined values (ALL-configured-address, ALL-physical-address and NO-BODY).

4.10.3.2.10 repeater

Holds the MIB variable *repeater* (variable 5) specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

It specifies whether the server system effectively repeats all frames or not.

enum:

(0) never repeater,
(1) always repeater,
(2) dynamic repeater

If the *repeater* variable is equal to 0, the server system should never repeat the frames.

If it is set to 1, the server system is a repeater: it has to repeat all frames received without error and with a current credit greater than zero.

If it is set to 2, then the repeater status can be dynamically changed by the server itself.

NOTE The value 2 value is not specified in IEC 61334-4-512.

This attribute is internally read by the MAC sub-layer each time a frame is received.

The default value shall be specified in project specific companion specifications.

4.10.3.2.11 repeater_status

Holds the current *repeater status* of the device.

boolean:

(0) FALSE = no repeater,
(1) TRUE = repeater

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2.12 min_delta_credit

Holds the MIB variable *min-delta-credit* (variable 9) specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

NOTE Only the three least significant bits are used.

The Delta Credit (DC) is the subtraction of the Initial Credit (IC) and Current Credit (CC) fields of a correct received MAC frame. The delta-credit minimum value of a correct received MAC frame, directed to a server system, is held by this variable.

The default value is set to the maximal initial credit (see IEC 61334-5-1:2001, 4.2.3.1 for further explanations on the credit and the value of MAX_INITIAL_CREDIT). A client system can reinitialise this variable by setting its value to the maximal initial credit.

4.10.3.2.13 initiator_mac_address

Holds the MIB variable *initiator-mac-address* specified in IEC 61334-5-1:2001, 4.3.7.6.

Its value is either the MAC address of the active-initiator or the NO-BODY address, depending on the value of the *synchronization_locked* attribute (see 4.10.3.2.14). See also IEC 61334-5-1:2001, 3.5.3, 4.1.6.3 and 4.1.7.2.

4.10.3.2.14 synchronization_locked

Holds the MIB variable *synchronization-locked* (variable 10) specified in IEC 61334-4-512:2001, 5.3.

Controls the synchronization locked / unlocked state. See IEC 61334-5-1:2001 for more details.

If the value of this attribute is equal to TRUE, the system is in the synchronization-locked state. In this state, the *initiator-mac-address* is always equal to the MAC address field of the active-initiator MIB object. See attribute 2 of the S-FSK Active initiator IC (4.10.4.2.2).

If the value of this attribute is equal to FALSE, the system is in the synchronization-unlocked state. In this state, the *initiator_mac_address* attribute is always set to the NO-BODY value: a value change in the MAC address field of the active-initiator MIB object does not affect the content of the *initiator_mac_address* attribute which remains at the NO-BODY value. The default value of this variable shall be specified in the implementation specifications.

NOTE In the synchronization-unlocked state, the server synchronizes on any valid frame. In the synchronization locked state, the server only synchronizes on frames issued or directed to the client system the MAC address of which is equal to the value of the initiator_mac_address attribute.

4.10.3.2.15 transmission_speed

The transmission speed supported by the physical device. See also IEC 61334-5-1:2001, 3.2.2.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

enum:	50 Hz	60 Hz
(0)	300 baud	360 baud
(1)	600 baud	720 baud
(2)	1 200 baud	1 440 baud
(3) -- default	2 400 baud	2 880 baud
(4)	4 800 baud	5 760 baud
(5)	7 200 baud	8 640 baud
(6)	9 600 baud	11 520 baud

Table 39 – MAC addresses in the S-FSK profile

Address	Value
NO-BODY	000
Local MAC	001...FIMA-1
Initiator	FIMA...LIMA
MAC group address	LIMA + 1...FFB
All configured	FFC
NEW	FFE
All Physical	FFF
NOTE MAC addresses are expressed on 12 bits. These addresses are specified in IEC 61334-5-1:2001, 4.2.3.2, 4.3.7.5.1, 4.3.7.5.2 and 4.3.7.5.3. FIMA: First Initiator MAC address; C00. LIMA: Last Initiator MAC address; DFF.	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.4 S-FSK Active initiator (class_id = 51, version = 0)

4.10.4.1 Overview

An instance of the “S-FSK Active initiator” IC stores the data of the active initiator. The active initiator is the client system, which has last registered the server system with a CIASE Register request. See IEC 61334-4-511:2000, 7.2.

S-FSK Active initiator		0...n	class_id = 51, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. active_initiator	(dyn.)	initiator_descriptor				x + 0x08
Specific methods		***m/o***				
1. reset_NEW_not_synchronized (data)						x + 0x10

4.10.4.2 Attribute description

4.10.4.2.1 logical_name

Identifies the “S-FSK Active initiator” object instance. See 6.2.25.

4.10.4.2.2 active_initiator

Holds the MIB variable *active-initiator* (variable 15) specified in IEC 61334-4-512:2001, 5.6.

Contains the identifiers of the active initiator, which has last registered the system with a Register request. See IEC 61334-4-511:2000, 7.2.

The Initiator system is identified with its System Title, MAC address and L-SAP selector:

```
initiator_descriptor::= structure

{
    system_title:       octet-string,
    MAC_address:        long-unsigned,
    L_SAP_selector:     unsigned
}
```

The size and the structure of the system title may be specified in system specifications. When the system title is used as part of the initialisation vector of cryptographic algorithms, then the size shall meet the requirements applicable for the initialisation vector.

The MAC_address element is used to update the *initiator-mac-address* MAC management variable when the system is configured in the synchronization-locked state. See the specification of the *initiator_mac_address* and the *synchronization_locked* attributes of the S-FSK Phy&MAC setup IC in 4.10.3.

As long as the server is not registered by an active initiator, the L_SAP_selector field is set to 0 and the system_title field is equal to an octet string of 0s.

The default value of the initiator-descriptor is: system_title = octet-string of 0s, MAC_address = NO-BODY and L_SAP_selector = 0.

The value of this attribute can be updated by the invocation of the *reset_NEW_not_synchronized* method or by the CIASE Register service.

4.10.4.3 Method description

4.10.4.3.1 reset_NEW_not_synchronized (data)

Holds the MIB variable *reset-NEW-not-synchronized* (variable 17) specified in IEC 61334-4-512:2001, 5.8.

Allows a client system to “reset” the server system. The submitted value corresponds to a client MAC address. The writing is refused if:

– the value does not correspond to a valid client MAC address or the predefined NO-BODY address;

– the submitted value is different from the NO-BODY address and the *synchronization_locked* attribute is not equal to TRUE.

For the description of the Intelligent Search Initiator process, see IEC 62056-8-3:2013, 10.7.

When this method is invoked, the following actions are performed:

– the system returns to the unconfigured state (UNC: MAC-address equals NEW-address). This transition automatically causes the synchronization lost (function of the MAC sub layer);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– the system changes the value of the *active_initiator* attribute: the MAC_address is set to the submitted value, the L-SAP_selector is set to the value 0 and the system_title is set to an octet-string of 0s;

– all AAs that can be released are released.

4.10.5 S-FSK MAC synchronization timeouts (class_id = 52, version = 0)

4.10.5.1 Overview

An instance of the "S-FSK MAC synchronization timeouts" IC stores the timeouts related to the synchronization process.

S-FSK MAC synchronization timeouts		**0...n**	**class_id = 52, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. search_initiator_timeout	(static)	long-unsigned				x + 0x08
3. synchronization_confirmation_ timeout	(static)	long-unsigned				x + 0x10
4. time_out_not_addressed	(static)	long-unsigned				x + 0x18
5. time_out_frame_not_OK	(static)	long-unsigned				x + 0x20
Specific methods		***m/o***				

4.10.5.2 Attribute description

4.10.5.2.1 logical_name

Identifies the "S-FSK synchronization timeouts" object instance. See 6.2.25.

4.10.5.2.2 search_initiator_timeout

This timeout supports the intelligent search initiator function.

It defines the value of the time, expressed in seconds, during which the server system is searching for the initiator with the strongest signal.

During this timeout, all initiators, which may be heard by the servers, are expected to talk.

After the expiry of this timeout, the server will accept a Register request from the initiator having provided the strongest signal and it will be locked to that initiator.

If the value of the timeout is equal to 0, this means that the feature is not used.

The timeout is started at the beginning of the Search Initiator Phase, when the server receives the first frame with a valid initiator MAC address. The timeout is restarted when the Search Initiator Phase is over and the server locks on the initiator. During the Check Initiator Phase, it is restarted on the reception of each valid frame.

A Fast synchronization may be performed if the level of signal is good enough (Level of initiator signal >= Search-Initiator-Threshold) and one of the MAC addresses (Source or Destination) is an Initiator MAC address. This means that the module (the meter) is next to a DC or next to a module that is already locked on that DC. The module locks in this case on that initiator.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.5.2.3 synchronization_confirmation_timeout

Holds the MIB variable *synchronization-confirmation-timeout* (variable 6) specified in IEC 61334-4-512:2001, 5.3 and IEC 61334-5-1:2001, 4.3.7.6.

Defines the value of the time, expressed in seconds, after which a server system which just gets frame synchronized (detection of a data path equal to AAAA54C7 hex) will automatically lose its frame synchronization if the MAC sublayer does not identify a valid MAC frame. The timeout starts after the reception of the first four bytes of a physical frame.

The value of this variable can be modified by a client system. This time-out ensures a fast desynchronization of a system, which has synchronized on a wrong physical frame. See IEC 61334-5-1:2001, 3.5.3 for more details.

The default value of this variable should be specified in the implementation specifications.

A value equal to 0 is equivalent to cancel the use of the related *synchronization_confirmation_timeout* counter.

4.10.5.2.4 time_out_not_addressed

Holds the MIB variable *time-out-not-addressed* (variable 7) specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

Defines the time, in minutes, after which a server system that has not been individually addressed:

– returns to the non configured state (UNC: MAC-address equals NEW-address): this transition automatically involves the loss of the synchronization (function of the MAC sub layer) and releasing all AAs that can be released;

– loses its active initiator: the MAC address of the active-initiator is set to NO-BODY, the LSAP selector is set to the value 00 and the System Title is set to an octet-string of 0s.

Because broadcast addresses are not individual system addresses, the timer associated with the *time-out-not-addressed* delay ensures that a forgotten system will sooner or later return to the unconfigured state. It will be then discovered again.

A forgotten system is a system, which has not been individually addressed for more than the "*time-out-not-addressed*" amount of time.

The default value of this variable should be specified in the implementation specifications.

A value equal to 0 is equivalent to cancel the use of the related time-out-not-addressed counter.

4.10.5.2.5 time_out_frame_not_OK

Holds the MIB variable *time-out-frame-not-OK* (variable 8), specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

Defines the time, in seconds, after which a server system that has not received a properly formed MAC frame (incorrect NS field, inconsistent number of received sub frames, false Cyclic Redundancy Code checking) loses its frame synchronization.

The default value of this variable shall be specified in the implementation specifications.

A value equal to 0 is equivalent to cancel the use of the related *time-out-frame-not-OK* counter.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.6 S-FSK MAC counters (class_id = 53, version = 0)

4.10.6.1 Overview

An instance of the “S-FSK MAC counters” IC stores counters related to the frame exchange, transmission and repetition phases.

S-FSK MAC counters		0...n	class_id = 53, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. synchronization_register	(dyn.)	array				x + 0x08
3. desynchronization_listing	(dyn.)	structure				x + 0x10
4. broadcast_frames_counter	(dyn.)	array				x + 0x18
5. repetitions_counter	(dyn.)	double-long-unsigned			0	x + 0x20
6. transmissions_counter	(dyn.)	double-long-unsigned			0	x + 0x28
7. CRC_OK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x30
8. CRC_NOK_frames_counter	(dyn.)	double-long-unsigned				x + 0x38
Specific methods		***m/o***				
1. reset (data)						x + 0x50

4.10.6.2 Attribute description

4.10.6.2.1 logical_name

Identifies the “S-FSK MAC counters” object instance. See 6.2.25.

4.10.6.2.2 synchronization_register

Holds the MIB variable *synchronization-register* (variable 23), specified in IEC 61334-4-512:2001, 5.8.

```
array synchronization_couples

synchronization_couples::= structure
{
        mac_address:                long-unsigned,
        synchronizations_counter:   double-long-unsigned
}--
```

This variable counts the number of synchronization processes performed by the system. Processes that lead to a synchronization loss due to the detection of a wrong initiator are registered. The other processes that lead to a synchronization loss (time-out, management writing) are not registered.

This variable provides a balance sheet of the different systems on which the server system is "potentially" able to synchronize.

A synchronization process is initialized when the Management Application Entity (connection manager) receives a MA_Sync.indication (Synchronization State = SYNCHRO_FOUND) primitive from the MAC Sublayer Entity. This process is registered in the synchronization-register variable only if the MA_Sync.indication (Synchronization State = SYNCHRO_FOUND) primitive is followed by one of the three primitives:

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

1) MA_Data.indication (DA, SA, MSDU) primitive;
2) MA_Sync.indication (Synchronization State = SYNCHRO_CONF, SA, DA);
3) MA_Sync.indication (Synchronization State = SYNCHRO_LOSS, Synchro Loss Cause = wrong_initiator, SA, DA

NOTE The third primitive is only generated if the server system is configured in a synchronization-locked state. See 4.10.3.

Processes which lead to the generation of MA_Sync.indication (Synchronization State = SYNCHRO_LOSS) primitives indicating synchronization loss due to:

– the physical layer;
– the time-out-not-addressed counter;
– setting the mac_address attribute of the S-FSK Phy&MAC setup object to NEW; see 4.10.3; or
– invoking the reset_NEW_not_synchronized method of the S-FSK Active initiator object; see 4.10.4 (this is known as Management Writing)

are not taken into account in this variable.

For details on the MA_Sync.indication service primitive, see IEC 61334-5-1:2001, 4.1.7.1.

If the synchronization process ends with one of the three primitives listed above, the synchronization-register variable is updated by taking into account the SA and DA fields of the primitive.

The updating of the *synchronization-register* variable is carried out as follows:

First, the Management Entity checks the SA and DA fields.

– If one of these fields corresponds to a client MAC address (CMA) the Entity:
  - checks if the client MAC address (CMA) appears in one of the couples contained in the synchronization-register variable;
  - if it appears, the related synchronizations-counter subfield is incremented;
  - if it does not appear, a new (mac-address, synchronizations-counter) couple is added. This couple is initialized to the (CMA, 1) value.
– If none of the SA and DA fields correspond to a client MAC address, it is supposed that the system found its synchronization reference on a DiscoverReport type frame. In that case, the mac-address which should be registered in the synchronization-register variable is the predefined NEW value (FFE). The updating of the synchronization-register variable is carried out in the same way as it is done for a normal client MAC address (CMA).

When a *synchronizations-counter* field reaches the maximum value, it automatically returns to 0 on the next increment.

The maximum number of synchronization couples {mac-address, synchronizations-counter} contained in this variable should be specified in the implementation specifications. When this maximum is reached, the updating of the variable follows a First-In-First-Out (FIFO) mechanism: only the newest source MAC addresses are memorized.

The default value of this variable is an empty array.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.6.2.3 desynchronization_listing

Holds the MIB variable *desynchronization-listing* (variable 24), specified in IEC 61334-4-512:2001, 5.8.

```
structure

{
        nb_physical_layer_desynchronization:                double-long-unsigned,
        nb_time_out_not_addressed_desynchronization:        double-long-unsigned,
        nb_timeout_frame_not_OK desynchronization:          double-long-unsigned,
        nb_write_request_desynchronization:                 double-long-unsigned,
        nb_wrong_initiator_desynchronization:               double-long-unsigned
}
```

This variable counts the number of desynchronizations that occurred depending on their cause. On reception of synchronization loss notification, the Management Entity updates this attribute by incrementing the counter related to the cause of the desynchronization.

When one of the counters reaches the maximum value, it automatically returns to 0 on the next increment.

The default value of this variable is a structure with all elements equal to 0.

4.10.6.2.4 broadcast_frames_counter

Holds the MIB variable *broadcast-frames-counter* (variable 19) specified in IEC 61334-4-512:2001, 5.8.

```
array       broadcast-couples

broadcast-couples::= structure

{
        source-mac-address:   long-unsigned,
        frames-counter:       double-long-unsigned
}
```

It counts the broadcast frames received by the server system and issued from a client system (source-mac-address = any valid client-mac-address, destination-mac address = ALL-physical). The number of frames is classified according to the origin of the transmitter. The counter is incremented even if the LLC-destination-address is not valid on the server system. When the frames-counter field reaches its maximum value, it automatically returns to 0 on the next increment.

The maximum number of broadcast-couples {source-mac-address, frames-counter} contained in this variable should be specified in the implementation specifications. When this maximum is reached, the updating of the variable follows a First-In-First-Out (FIFO) mechanism: only the newest source MAC addresses are memorized.

4.10.6.2.5 repetitions_counter

Holds the MIB variable *repetitions-counter* (variable 20) specified in IEC 61334-4-512:2001, 5.8.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Counts the number of repetition phases. The repetition phases following a transmission are not counted. If the MAC sub-layer is configured in the no-repeater mode, this variable is not updated. The repetitions counter measures the activity of the system as a repeater. A received frame repeated five times (from CC=4 to CC=0) is counted only once in the repetitions counter since it corresponds to one repetition phase. The counter is incremented at the beginning of each repetition phase. When the repetitions counter reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

4.10.6.2.6 transmissions_counter

Holds the MIB variable *transmissions-counter* (variable 21) specified in IEC 61334-4-512:2001, 5.8.

Counts the number of transmission phases. A transmission phase is characterized by the transmission and the repetition of a frame. A repetition phase, which follows the reception of a frame, is not counted. The transmission counter is incremented at the beginning of each transmission phase. A client system can write this variable to update the counter. When the transmissions counter reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

4.10.6.2.7 CRC_OK_frames_counter

Holds the MIB variable *CRC-OK-frames-counter* (variable 22) specified in IEC 61334-4-512:2001, 5.8

Counts the number of frames received with a correct Frame Check Sequence Field. When the CRC OK frames counter field reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

4.10.6.2.8 CRC_NOK_frames_counter

Counts the number of frames received with an incorrect Frame Check Sequence Field. When the CRC NOK frames counter field reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

4.10.6.3 Method description

4.10.6.3.1 reset (data)

Clears all counters.

data::= integer (0)

4.10.7 IEC 61334-4-32 LLC setup (class_id = 55, version = 1)

4.10.7.1 Overview

An instance of the “IEC 61334-4-32 LLC setup” IC holds parameters necessary to set up and manage the LLC layer as specified in IEC 61334-4-32.

IEC 61334-4-32 LLC setup		**0…n**	**class_id = 55, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. max_frame_length	(static)	long-unsigned				x + 0x08
3. reply_status_list	(dyn.)	array				x + 0x10
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.7.2 Attribute description

4.10.7.2.1 logical_name

Identifies the “IEC 61334-4-32 LLC setup” object instance. See 6.2.25.

4.10.7.2.2 max_frame_length

Holds the length of the LLC frame in bytes. See IEC 61334-4-32:1996, 5.1.4.

For the S-FSK PLC profile, the minimum / default / maximum values are 26 / 134 / 242 bytes respectively. See IEC 61334-5-1:2001, 4.2.2.

NOTE For other lower layer profiles, see the corresponding values in the relevant specification.

4.10.7.2.3 reply_status_list

Holds the MIB variable *reply-status-list* (variable 11) specified in IEC 61334-4-512:2001, 5.4.

Lists the L-SAPs that have a not empty RDR (Reply Data on Request) buffer, which has not already been read. The length of a waiting L-SDU is specified in number of sub frames (different from zero). The variable is locally generated by the LLC sub layer.

```
array        reply_status

reply_status::= structure

{
        L_SAP_selector:          unsigned,
        length_of_waiting_L_SDU: unsigned
}
```

length_of_waiting_L_SDU in the case of the S-FSK profile is in number of sub-frames; valid values are 1 to 7.

4.10.8 S-FSK Reporting system list (class_id = 56, version = 0)

4.10.8.1 Overview

An instance of the “S-FSK Reporting system list” IC holds the list of reporting systems.

S-FSK Reporting system list		**0…n**	**class_id = 56, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. reporting_system_list	(dyn.)	array				x + 0x08
Specific methods		***m/o***				

4.10.8.2 Attribute description

4.10.8.2.1 logical_name

Identifies the “S-FSK Reporting system list” object instance. See 6.2.25.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.8.2.2 reporting_system_list

Holds the MIB variable *reporting-system-list* (variable 16) specified in IEC 61334-4-512:2001, 5.7.

array	system-title
system-title::=	octet-string

Contains the system-titles of the server systems which have made a DiscoverReport request and which have not already been registered. The list has a finite size and it is sorted upon the arrival. The first element is the newest one. Once full, the oldest ones are replaced by the new ones.

The reporting system list is updated:

– when a DiscoverReport CI_PDU is received by the server system (whatever its state: non configured or configured): the CIASE adds the reporting system-title at the beginning of the list, and verifies that it does not exist anywhere else in the list, if so it destroys the old one. A system-title can only be present once in the list;

– when a Register CI_PDU is received by the server system (whatever its state: non configured or configured): the CIASE checks the reporting-system list. If a system-title is present in the reporting-system-list and in the Register CI-PDU, the CIASE deletes the system-title in the reporting-system-list: this system is no more considered as a reporting system.

4.11 Interface classes for setting up the LLC layer for ISO/IEC 8802-2

4.11.1 General

This subclause specifies the ICs available for setting up the ISO/IEC 8802-2 LLC layer, used in some DLMS®/COSEM communication profiles, in the various types of operation.

For definitions related to the ISO/IEEE 8802-2 LLC layer see ISO/IEC 8802-2:1998, 1.4.2

4.11.2 ISO/IEC 8802-2 LLC Type 1 setup (class_id = 57, version = 0)

4.11.2.1 Overview

An instance of the “ISO/IEC 8802-2 LLC Type 1 setup” IC holds the parameters necessary to set up the ISO/IEC 8802-2 LLC layer in Type 1 operation.

ISO/IEC 8802-2 LLC Type 1 setup	**0…n**	**class_id = 57, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. max_octets_ui_pdu (static)	long unsigned			128	x + 0x08
Specific methods	***m/o***				

4.11.2.2 Attribute description

4.11.2.2.1 logical_name

Identifies the “ISO/IEC 8802-2 LLC Type 1 setup” object instance. See 6.2.26.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.11.2.2.2 max_octets_ui_pdu

Refer to the appropriate MAC protocol specification for any limitation on the maximum number of octets in a UI PDU. No restrictions are imposed by the LLC sublayer. However, in the interest of having a value that all users of Type 1 LLC may depend upon, all MACs shall at least be capable of accommodating UI PDUs with information fields up to and including 128 octets in length.

See ISO/IEC 8802-2:1998, 6.8.1 *Maximum number of octets* in a UI PDU.

4.11.3 ISO/IEC 8802-2 LLC Type 2 setup (class_id = 58, version = 0)

4.11.3.1 Overview

An instance of the “ISO/IEC 8802-2 LLC Type 2 setup” IC holds the parameters necessary to set up the ISO/IEC 8802-2 LLC layer in Type 2 operation.

ISO/IEC 8802-2 LLC Type 2 setup		**0…n**	**class_id = 58, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. transmit_window_size_k	(static)	unsigned	1	127	1	x + 0x08
3. receive_window_size_rw	(static)	unsigned	1	127	1	x + 0x10
4. max_octets_i_pdu_n1	(static)	long unsigned			128	x + 0x18
5. max_number_transmissions_n2	(static)	unsigned				x + 0x20
6. acknowledgement_timer	(static)	long-unsigned				x + 0x28
7. p_bit_timer	(static)	long-unsigned				x + 0x30
8. reject_timer	(static)	long-unsigned				x + 0x38
9. busy_state_timer	(static)	long-unsigned				x + 0x40
Specific methods		***m/o***				

4.11.3.2 Attribute description

4.11.3.2.1 logical_name

Identifies the “ISO/IEC 8802-2 LLC Type 2 setup” object instance. See 6.2.26.

transmit_window_size_k

The transmit window size (k) shall be a data link connection parameter that can never exceed 127. It shall denote the maximum number of sequentially numbered I PDUs that the sending LLC may have outstanding (i.e., unacknowledged). The value of k is the maximum number by which the sending LLC send state variable V(S) can exceed the N(R) of the last received I PDU.

See ISO/IEC 8802-2:1998, 7.8.4 *Transmit window size, k.*

4.11.3.2.2 receive_window_size_rw

The receive window size (RW) shall be a data link connection parameter that can never exceed 127. It shall denote the maximum number of unacknowledged sequentially numbered I PDUs that the local LLC allows the remote LLC to have outstanding. It is transmitted in the information field of XID (see ISO/IEC 8802-2:1998, 5.4.1.1.2) and applies to the XID sender. The XID receiver shall set its transmit window (k) to a value less than or equal to the receive window of the XID sender to avoid overrunning the XID sender.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

See ISO/IEC 8802-2:1998, 7.8.6 *Receive window size, RW*.

4.11.3.2.3 max_octets_i_pdu_n1

N1 is a data link connection parameter that denotes the maximum number of octets in an I PDU. Refer to the various MAC descriptions to determine the precise value of N1 for a given medium access method. LLC itself places no restrictions on the value of N1. However, in the interest of having a value of N1 that all users of Type 2 LLC may depend upon, all MACs shall at least be capable of accommodating I PDUs with information fields up to an including 128 octets in length.

See ISO/IEC 8802-2:1998, 7.8.3 *Maximum number of octets in an I PDU, N1.*

4.11.3.2.4 max_number_transmissions_n2

N2 is a data link connection parameter that indicates the maximum number of times that a PDU is sent following the running out of the acknowledgment timer, the P-bit timer, the reject timer, or the busy-state timer.

See ISO/IEC 8802-2:1998, 7.8.2 *Maximum number of transmissions, N2.*

4.11.3.2.5 acknowledgement_timer

The acknowledgment timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive an acknowledgment to one or more outstanding I PDUs or an expected response PDU to a sent unnumbered command PDU. The unit is seconds.

See ISO/IEC 8802-2:1998, 7.8.1.1 *Acknowledgement timer.*

4.11.3.2.6 p_bit_timer

The P-bit timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a PDU with the F bit set to “1” in response to a sent Type 2 command with the P bit set to “1”. The unit is seconds.

See ISO/IEC 8802-2:1998, 7.8.1.2 *P-bit timer.*

4.11.3.2.7 reject_timer

The reject timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a reply to a sent REJ PDU. The unit is seconds.

See ISO/IEC 8802-2:1998, 7.8.1.3 *Reject timer.*

4.11.3.2.8 busy_state_timer

The busy-state timer is a data link connection parameter that shall define the timer interval during which the LLC shall wait for an indication of the clearance of a busy condition at the other LLC. The unit is seconds.

See ISO/IEC 8802-2:1998, 7.8.1.4 *Busy-state timer*.

4.11.4 ISO/IEC 8802-2 LLC Type 3 setup (class_id = 59, version = 0)

4.11.4.1 Overview

An instance of the “ISO/IEC 8802-2 LLC Type 3 setup” IC holds the parameters necessary to set up the ISO/IEC 8802-2 LLC layer in Type 3 operation.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

ISO/IEC 8802-2 LLC Type 3 setup		0...n	class_id = 59, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				X
2. max_octets_acn_pdu_n3	(static)	long unsigned				x + 0x08
3. max_number_transmissions_n4	(static)	unsigned				x + 0x10
4. acknowledgement_time_t1	(static)	long unsigned				x + 0x18
5. receive_lifetime_var_t2	(static)	long unsigned				x + 0x20
6. transmit_lifetime_var_t3	(static)	long unsigned				x + 0x28
Specific methods		***m/o***				

4.11.4.2 Attribute description

4.11.4.2.1 logical_name

Identifies the “ISO/IEC 8802-2 LLC Type 3 setup” object instance. See 6.2.26.

4.11.4.2.2 max_octets_acn_pdu_n3

N3 is a logical link parameter that denotes the maximum number of octets in an ACn command PDU. Refer to the various MAC descriptions to determine the precise value of N3 for a given medium access method. LLC places no restrictions on the value of N3.

See ISO/IEC 8802-2:1998, 8.6.2 *Maximum number of octets in an ACn command PDU, N3.*

4.11.4.2.3 max_number_transmissions_n4

N4 is a logical link parameter that indicates the maximum number of times that an ACn command PDU is sent by LLC trying to accomplish a successful information exchange. Normally, N4 is set large enough to overcome the loss of a PDU due to link error conditions. If the medium access control sublayer has its own retransmission capability, the value of N4 may be set to one so that LLC does not itself requeue a PDU to the medium access control sublayer.

See ISO/IEC 8802-2:1998, 8.6.1 *Maximum number of transmissions, N4.*

4.11.4.2.4 acknowledgement_time_t1

The acknowledgment time is a logical link parameter that determines the period of the acknowledgment timers, and as such shall define the time interval during which the LLC shall expect to receive an ACn response PDU from a specific LLC from which the LLC is awaiting a response PDU. The acknowledgment time shall take into account any delay introduced by the MAC sublayer and whether the timer is started at the beginning or at the end of the sending of the ACn command PDU by the LLC. The proper operation of the procedure shall require that the acknowledgment time be greater than the normal time between the sending of an ACn command PDU and the reception of the corresponding ACn response PDU. If the medium access control sublayer performs its own retransmissions and if the logical link parameter N4 is set to one to prevent LLC from re-queuing a PDU, then the acknowledgment time T1 may be set to infinity, making the acknowledgment timers unnecessary.

The unit is seconds. Infinity is indicated by all bits set to 1.

See ISO/IEC 8802-2:1998, 8.6.4, *Acknowledgement time, T1.*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.11.4.2.5 receive_lifetime_var_t2

This time value is a logical link parameter that determines the period of all of the receive variable lifetime timers. T2 shall be longer by a margin of safety than the longest possible period during which the first transmission and all retries of a single PDU may occur. The margin of safety shall take into account anything affecting LLCs perception of the arrival time of PDUs, such as LLC response time, timer resolution, and variations in the time required for the medium access control sublayer to pass received PDUs to LLC.

If the destruction of the received state variables is not desired, the value of time T2 may be set to infinity. In this case the receive variable lifetime timer need not be implemented.

The unit is seconds. Infinity is indicated by all bits set to 1.

See ISO/IEC 8802-2:1998, 8.6.5 *Receive lifetime variable, T2.*

4.11.4.2.6 transmit_lifetime_var_t3

This time value is a logical link parameter that determines the minimum lifetime of the transmit sequence state variables. T3 shall be longer by a margin of safety than

a) the logical link variable T2 at stations to which ACn commands are sent; and

b) the longest possible lifetime of an ACn command-response pair. The lifetime of an ACn command-response pair shall take into account the sum of processing time, queuing delays, and transmission time for the command and response PDUs at the local and remote stations.

If the destruction of the transmit state variables is not desired, the value of time T3 may be set to infinity. Note, if the receive variable lifetime parameter, T2 is set to infinity at remote stations to which ACn commands are sent, then the T3 parameter shall be set to infinity at the local station.

The unit is seconds. Infinity is indicated by all bits set to 1.

See ISO/IEC 8802-2:1998, 8.6.6 *Transmit lifetime variable, T3.*

4.12 Interface classes for setting up and managing DLMS®/COSEM narrowband OFDM PLC profile for PRIME networks

4.12.1 Overview

See also Annex D.

COSEM objects for data exchange using narrowband OFDM PLC profile for PRIME networks, if implemented, shall be located in the Management Logical Device of COSEM servers.

Figure 28 shows an example with a COSEM physical device comprising three logical devices.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

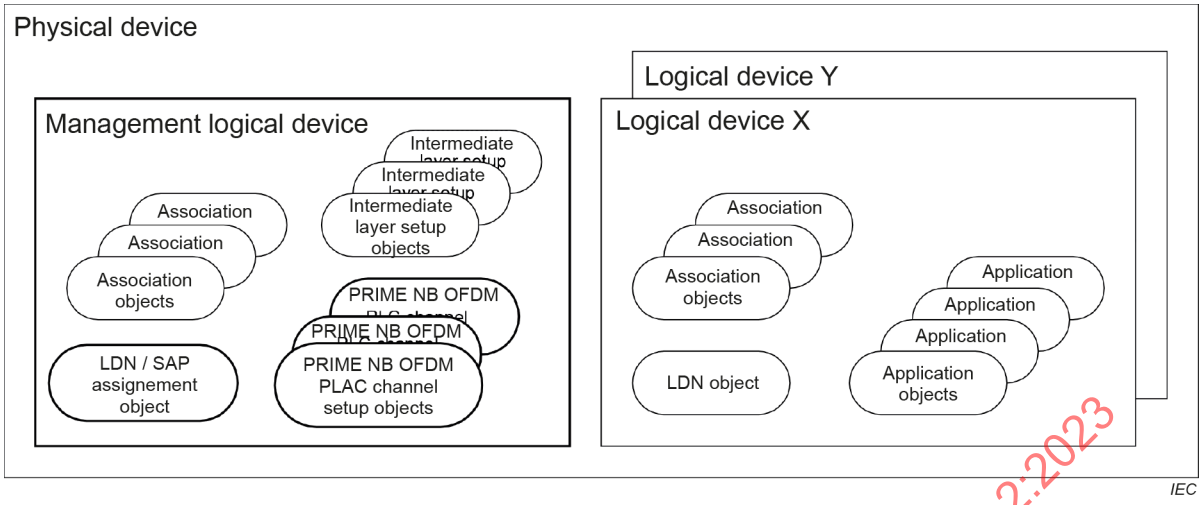


Figure 28 – Object model of DLMS®/COSEM servers

Each logical device shall contain a Logical Device Name (LDN) object.

NOTE As in this example there is more than one logical device, the mandatory Management logical device contains a "SAP Assignment" object instead of a Logical Device object.

Each logical device contains one or more "Association" objects, one for each client supported.

The management logical device contains the setup objects of the physical and MAC layers of narrowband OFDM PLC profile for PRIME networks as well as setup objects for the intermediate layer(s). It may contain further application objects.

The other logical devices, in addition to the "Association" and Logical Device Name objects mentioned above, contain further application objects, holding parameters and measurement values.

To set up and manage the 61334-4-32 LLC SSCS, one IC is specified:

- "61334-4-32 LLC SSCS setup", see 4.12.3;

To manage the PRIME NB OFDM PLC physical layer (PhL), one IC is specified:

- "PRIME NB OFDM PLC Physical layer counters", see 4.12.5;

To set up and manage the PLC PRIME OFDM MAC layer, four ICs are specified:

- "PRIME NB OFDM PLC MAC setup": see 4.12.6;
- "PRIME NB OFDM PLC MAC functional parameters": see 4.12.7;
- "PRIME NB OFDM PLC MAC counters": see 4.12.8;
- "PRIME NB OFDM PLC MAC network administration data": see 4.12.9.

For application identification, one IC is specified:

- "PRIME NB OFDM PLC Application identification", see 4.12.11.

4.12.2 Mapping of PRIME NB OFDM PLC PIB attributes to COSEM IC attributes

ITU-T G.9904:2012 defines variables in Table 10-1 and Table 10-2 for PHY PIB attributes Table 10-3 to Table 10-8 for MAC PIB attributes and Table 10-9 for Applications PIB attributes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 40 shows the mapping of PRIME NB OFDM PLC PIB attributes to attributes of COSEM ICs. Only variables related to the switch and Terminal nodes are mapped. Variables relevant for the base node are not mapped, because the base node acts as a client regarding the distribution network.

Table 40 – Mapping of PRIME NB OFDM PLC PIB attributes to COSEM IC attributes

Name	Identifier	Interface class	class_id / attribute
PHY PIB attributes – PHY read-only variable that provide statistical information [1]			
phyStatsCRCIncorrectCount	0x00A0	PRIME NB OFDM PLC Physical layer counters (class_id = 81, version = 0)	81 / Attr. 2
PhyStatsCRCFailCount	0x00A1		81 / Attr. 3
phyStatsTxDropCount	0x00A2		81 / Attr. 4
phyStatsRxDropCount	0x00A3		81 / Attr. 5
phyStatsRxTotalCount	0x00A4		Not modelled
phyStatsBlkAvgEvm	0x00A5		Not modelled
phyEmaSmoothing	0x00A8		Not modelled
PHY read-only parameters, providing information on specific implementation [2]			
phyTxQueueLen	0x00B0		Not modelled
phyRxQueueLen	0x00B1		
phyTxProcessingDelay	0x00B2		
phyRxProcessingDelay	0x00B3		
PhyAgcMinGain	0x00B4		
PhyAgcStepValue	0x00B5		
PhyAgcStepNumber	0x00B6		
MAC read-write variables, read-only variables [3]			
macMinSwitchSearchTime	0x0010		82 / Attr. 2
macMaxPromotionPdu	0x0011	PRIME NB OFDM PLC MAC setup (class_id = 82, version = 0)	82 / Attr. 3
macMaxPromotionPduTxPeriod	0x0012		82 / Attr. 4
macBeaconsPerFrame	0x0013		82 / Attr. 5
macSCPMaxTxAttempts	0x0014		82 / Attr. 6
macCtlReTxTimer	0x0015		82 / Attr. 7
macMaxCtlReTx	0x0018		82 / Attr. 8
macEMASmoothing	0x0019		Not modelled
macSCPRBO	0x0016		Not modelled
macSCPChSenseCount	0x0017		Not modelled
MAC read-only variables that provide functional information [4]			
macLNID	0x0020	NB OFDM PLC MAC functional parameters (class_id = 83 version = 0)	83 / Attr. 2
macLSID	0x0021		83 / Attr. 3
macSID	0x0022		83 / Attr. 4
macSNA	0x0023		83 / Attr. 5
macState	0x0024		83 / Attr. 6
macSCPLength	0x0025		83 / Attr. 7
macNodeHierarchyLevel	0x0026		83 / Attr. 8
macBeaconSlotCount	0x0027		83 / Attr. 9
macBeaconRxSlot	0x0028		83 / Attr. 10
macBeaconTxSlot	0x0029		83 / Attr. 11

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Name	Identifier	Interface class	class_id / attribute
macBeaconRxFrequency	0x002A		83 / Attr. 12
macBeaconTxFrequency	0x002B		83 / Attr. 13
macCapabilities	0x002C		83/ Attr. 14
MAC read-only variable that provide statistical information [5]			
macTxDataPktCount	0x0040	PRIME NB OFDM PLC MAC counters (class_id = 84, version = 0)	84 / Attr. 2
macRxDataPktCount	0x0041		84 / Attr. 3
macTxCtrlPktCount	0x0042		84 / Attr. 4
macRxCtrlPktCount	0x0043		84 / Attr. 5
macCSMAFailCount	0x0044		84 / Attr. 6
macCSMAChBusyCount	0x0045		84 / Attr. 7
Read-only lists, made available by MAC layer through management interface [6]			
macListMcastEntries	0x0052	PRIME NB OFDM PLC MAC network administration data (class_id = 85, version = 0)	85 / Attr. 2
macListSwitchTable	0x0053		85 / Attr. 3
macListDirectTable	0x0055		85 / Attr. 4
macListAvailableSwitches	0x0056		85 / Attr. 5
macListPhyComm	0x0057		85 / Attr. 6
Application PIB attributes [7]			
AppFwVersion	0x0075	PRIME NB OFDM PLC Application identification (class_id = 86, version = 0)	86 / Attr. 2
AppVendorId	0x0076		86 / Attr. 3
AppProductId	0x0077		86 / Attr. 4

[1] See ITU-T G.9904:2012, Table 10-1.

[2] See ITU-T G.9904:2012, Table 10-2.

[3] See ITU-T G.9904:2012, Table 10-3, 10-4.

[4] See ITU-T G.9904:2012, Table 10-5.

[5] See ITU-T G.9904:2012, Table 10-6.

[6] See ITU-T G.9904:2012, Table 10-7.

[7] See ITU-T G.9904:2012, Table 10-9.

NOTE Whereas in COSEM interface class specifications the underscore notation is used, in Recommendation ITU-T G.9904:2012 and in Table 1, the camel notation is used.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.3 61334-4-32 LLC SSCS setup (class_id = 80, version = 0)

4.12.3.1 Overview

An instance of the “61334-4-32 LLC SSCS” (Service Specific Convergence Sublayer, 432 CL) setup IC holds addresses that are provided by the base node during the opening of the convergence layer, as a response to the establish request of the service node. They allow the service node to be part of the network managed by the base node.

61334-4-32 LLC SSCS setup	0...n	class_id = 80, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. service_node_address (dyn.)	long-unsigned				x + 0x08
3. base_node_address (dyn.)	long-unsigned				x + 0x10
Specific methods	***m/o***				
1. reset (data)	o				x + 0x20

4.12.3.2 Attribute description

4.12.3.2.1 logical_name

Identifies the “61334-4-32 LLC SSCS setup object instance”. See 6.2.27.

4.12.3.2.2 service_node_address

Holds the value of the address assigned to the service node during its registration by the base node.

After deregistration, the value of this address is NEW, meaning 0xFFE.

4.12.3.2.3 base_node_address

Holds the value of the base node address to which the service node is registered.

After deregistration this address is 0.

4.12.3.3 Method description

4.12.3.3.1 reset (data)

This method is used for deallocating the service node address.

The value of the *service_node_address* becomes NEW and the value of the *base_node_address* becomes 0.

4.12.4 PRIME NB OFDM PLC Physical layer parameters

The physical layer parameters are not modelled.

4.12.5 PRIME NB OFDM PLC Physical layer counters (class_id = 81, version = 0)

4.12.5.1 Overview

An instance of the “PRIME NB OFDM PLC Physical layer counters” IC stores counters related to the physical layers exchanges. The objective of these counters is to provide statistical information for management purposes.

The attributes of instances of this IC shall be read only. They can be reset using the reset method.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PRIME NB OFDM PLC Physical layer counters	0...n	class_id = 81, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. phy_stats_crc_incorrect_count (dyn.)	long-unsigned				x + 0x08
3. phy_stats_crc_fail_count (dyn.)	long-unsigned				x + 0x10
4. phy_stats_tx_drop_count (dyn.)	long-unsigned				x + 0x18
5. phy_stats_rx_drop_count (dyn.)	long-unsigned				x + 0x20
Specific methods	***m/o***				
1. reset (data)	o				

4.12.5.2 Attribute description

4.12.5.2.1 General

logical_name

Identifies the “PRIME NB OFDM PLC Physical layer counters” object instance. See 6.2.27.

4.12.5.2.2 phy_stats_crc_incorrect_count

PIB attribute 0x00A0: Number of bursts received on the physical layer for which the CRC was incorrect.

4.12.5.2.3 phy_stats_crc_failed_count

PIB attribute 0x00A1: Number of bursts received on the physical layer for which the CRC was correct, but the Protocol field of PHY header had invalid value. This count would reflect number of times corrupt data was received and the CRC calculation failed to detect it.

4.12.5.2.4 phy_stats_tx_drop_count

PIB attribute 0x00A2: Number of times when PHY layer received new data to transmit (PHY_DATA.request) and had to either overwrite on existing data in its transmit queue or drop the data in new request due to full queue.

4.12.5.2.5 phy_stats_rx_drop_count

PIB attribute 0x00A3: Number of times when the PHY layer received new data on the channel and had to either overwrite on existing data in its receive queue or drop the newly received data due to full queue.

NOTE When a counter reaches the maximum value (0xFFFF), it is automatically rolled-over.

4.12.5.3 Method description

4.12.5.3.1 reset (data)

This method is used for resetting all the counters held by an instance of this interface.

4.12.6 PRIME NB OFDM PLC MAC setup (class_id = 82, version = 0)

4.12.6.1 Overview

An instance of the “PRIME NB OFDM PLC MAC setup” IC holds the necessary parameters to set up and manage the PRIME NB OFDM PLC MAC layer.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

These attributes influence the functional behaviour of an implementation. These attributes may be defined external to the MAC, typically by the management entity and implementations may allow changes to their values during normal running, i.e. even after the device start-up sequence has been executed.

NOTE When a counter reaches the maximum value (0xFFFF), it is automatically rolled-over.

PRIME NB OFDM PLC MAC setup		**0...n**	**class_id = 82, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mac_min_switch_search_time	(static)	unsigned	16	32	24	x + 0x08
3. mac_max_promotion_pdu	(static)	unsigned	1	4	2	x + 0x10
4. mac_promotion_pdu_tx_period	(static)	unsigned	2	8	5	x + 0x18
5. mac_beacons_per_frame	(static)	unsigned	1	5	5	x + 0x20
6. mac_scp_max_tx_attempts	(static)	unsigned	2	5	5	x +0x28
7. mac_ctl_re_tx_timer	(static)	unsigned	2	20	15	x + 0x30
8. mac_max_ctl_re_tx	(static)	unsigned	3	5	3	x + 0x38
Specific methods		***m/o***				

4.12.6.2 Attribute description

4.12.6.2.1 logical_name

Identifies the “PRIME NB OFDM PLC MAC setup” object instance. See 6.2.27.

4.12.6.2.2 mac_min_switch_search_time

PIB attribute 0x0010: Minimum time for which a service node in *Disconnected* status should scan the channel for beacons before it can broadcast PNPDU. This attribute is not maintained in base nodes.

The unit of this attribute is seconds.

4.12.6.2.3 mac_max_promotion_pdu

PIB attribute 0x0011: Maximum number of PNPDUs that may be transmitted by a service node in a period of *mac_promotion_pdu_tx_period* seconds. This attribute is not maintained in base nodes.

4.12.6.2.4 mac_promotion_pdu_tx_period

PIB attribute 0x0012: Time quantum for limiting the number of PNDPUs transmitted from a service node. No more than *mac_max_promotion_pdu* may be transmitted in a period of *mac_promotion_pdu_tx_period*.

The unit of this attribute is seconds.

4.12.6.2.5 mac_beacons_per_frame

PIB attribute 0x0013: Maximum number of beacon slots that may be provisioned in a frame. This attribute is maintained in base nodes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.6.2.6 mac_scp_max_tx_attempts

PIB attribute 0x0014: Number of times the CSMA algorithm would attempt to transmit requested data when a previous attempt was withheld due to PHY indicating channel busy.

4.12.6.2.7 mac_ctl_re_tx_timer

PIB attribute 0x0015: Number of seconds for which a MAC entity waits for acknowledgement of receipt of MAC control packet from its peer entity. On expiry of this time, the MAC entity may retransmit the MAC control packet.

The unit of this attribute is seconds.

4.12.6.2.8 mac_max_ctl_re_tx

PIB attribute 0x0018: Maximum number of times a MAC entity will try to retransmit an unacknowledged MAC control packet. If the retransmit count reaches this maximum, the MAC entity shall abort further attempts to transmit the MAC control packet.

NOTE When a counter reaches the maximum value (0xFFFF), it is automatically rolled-over.

4.12.7 NB OFDM PLC MAC functional parameters (class_id = 83 version = 0)

4.12.7.1 Overview

The attributes of an instance of the “PRIME NB OFDM PLC MAC functional parameters” IC belong to the functional behaviour of MAC. They provide information on specific aspects.

The attributes of instances of this IC shall be read only.

PRIME NB OFDM PLC MAC functional parameters		**0...n**	**class_id = 83, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mac_LNID	(static)	long	0	16 383		x + 0x08
3. mac_LSID	(static)	unsigned	0	255		x + 0x10
4. mac_SID	(static)	unsigned	0	255		x + 0x18
5. mac_SNA	(static)	octet-string				x + 0x20
6. mac_state	(static)	enum	0	3		x +0x28
7. mac_scp_length	(static)	long				x + 0x30
8. mac_node_hierarchy_level	(static)	unsigned	0	63		x + 0x38
9. mac_beacon_slot_count	(static)	unsigned	0	7		x + 0x40
10. mac_beacon_rx_slot	(static)	unsigned	0	7		x + 0x48
11. mac_beacon_tx_slot	(static)	unsigned	0	7		x + 0x50
12. mac_beacon_rx_frequency	(static)	unsigned	0	31		x + 0x58
13. mac_beacon_tx_frequency	(static)	unsigned	0	31		x + 0x60
14. mac_capabilities	(static)	long-unsigned				x + 0x68
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.7.2 Attribute description

4.12.7.2.1 logical_name

Identifies the “PRIME NB OFDM PLC MAC functional parameters” object instance. See 6.2.27.

4.12.7.2.2 mac_LNID

PIB attribute 0x0020: LNID allocated to this node at time of its registration.

4.12.7.2.3 mac_LSID

PIB attribute 0x0021: LSID allocated to this node at the time of its promotion. This attribute is not maintained if the node is in a Terminal functional state.

4.12.7.2.4 mac_SID

PIB attribute 0x0022: SID of the switch node through which this node is connected to the subnetwork. This attribute is not maintained in a base node.

4.12.7.2.5 mac_SNA

PIB attribute 0x0023: Subnetwork address to which this node is registered. The base node returns the SNA it is using.

4.12.7.2.6 mac_state

PIB attribute 0x0024: Present functional state of the node.

enum:

(0) Disconnected,
(1) Terminal,
(2) Switch,
(3) Base

4.12.7.2.7 mac_scp_length

PIB attribute 0x0025: The SCP length, in symbols, in present frame.

4.12.7.2.8 mac_node_hierarchy_level

PIB attribute 0x0026: Level of this node in subnetwork hierarchy.

4.12.7.2.9 mac_beacon_slot_count

PIB attribute 0x0027: Number of beacon slots provisioned in present frame structure.

4.12.7.2.10 mac_beacon_rx_slot

PIB attribute 0x0028: Beacon slot in which this device’s switch node transmits its beacon. This attribute is not maintained in a base node.

4.12.7.2.11 mac_beacon_tx_slot

PIB attribute 0x0029: Beacon slot in which this device transmits its beacon. This attribute is not maintained in service nodes that are in a *Terminal* functional state.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.7.2.12 mac_beacon_rx_frequency

PIB attribute 0x002A: Number of frames between receptions of two successive beacons. A value of 0x0 indicates beacons are received in every frame. This attribute is not maintained in a base node.

4.12.7.2.13 mac_beacon_tx_frequency

PIB attribute 0x002B: Number of frames between transmissions of two successive beacons. A value of 0x0 indicates beacons are transmitted in every frame. This attribute is not maintained in service nodes that are in a *Terminal* functional state.

4.12.7.2.14 mac_capabilities

PIB attribute 0x002C: This attribute defines the capabilities of the node. It is a bitmap each bit defining a capability.

Bit 0: Switch Capable
Bit 1: Packet Aggregation
Bit 2: Contention Free Period
Bit 3: Direct connection
Bit 4: Multicast
Bit 5: PHY Robustness Management
Bit 6: ARQ
Bit 7; Reserved for future use
Bit 8: Direct Connection Switching
Bit 9: Multicast Switching Capability
Bit 10: PHY Robustness Management Switching Capability
Bit 11: ARQ Buffering Switching Capability
Bit 12 to 15: Reserved for future use

4.12.8 PRIME NB OFDM PLC MAC counters (class_id = 84, version = 0)

4.12.8.1 Overview

An instance of the “PRIME NB OFDM PLC MAC counters” IC stores statistical information on the operation of the MAC layer for management purposes. The attributes of instances of this IC shall be read only. They can be reset using the reset method.

NOTE When a counter reaches the maximum value (0xFFFF), it is automatically rolled-over.

PRIME NB OFDM PLC MAC counters	**0...n**	**class_id = 84, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. mac_tx_data_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x08
3. mac_rx_data_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x10
4. mac_tx_ctrl_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x18
5. mac_rx_ctrl_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x20
6. mac_csma_fail_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x28
7. mac_csma_ch_busy_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x30
Specific methods	***m/o***				
1. reset (data)	o				x + 0x40

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.8.2 Attribute description

4.12.8.2.1 logical_name

Identifies the “PRIME NB OFDM PLC MAC counters” object instance. See 6.2.27.

4.12.8.2.2 mac_tx_data_pkt_count

PIB attribute 0x0040: Count of successfully transmitted MSDUs.

4.12.8.2.3 mac_rx_data_pkt_count

PIB attribute 0x0041: Count of successfully received MSDUs whose destination address was this node.

4.12.8.2.4 mac_tx_ctrl_pkt_count

PIB attribute 0x0042: Count of successfully transmitted MAC control packets.

4.12.8.2.5 mac_rx_ctrl_pkt_count

PIB attribute 0x0043: Count of successfully received MAC control packets whose destination was this node.

4.12.8.2.6 mac_csma_fail_count

PIB attribute 0x0044: Count of failed CSMA transmit attempts.

4.12.8.2.7 mac_csma_ch_busy_count

PIB attribute 0x0045: Count of number of times this node has to back off SCP transmission due to channel busy state.

NOTE When a counter reaches the maximum value (0xFFFFFFFF), it is automatically rolled-over.

4.12.8.3 Method description

4.12.8.3.1 reset (data)

This method is used for resetting all the counters held by an instance of this interface class.

data::= integer (0)

4.12.9 PRIME NB OFDM PLC MAC network administration data (class_id = 85, version = 0)

4.12.9.1 Overview

This IC holds the parameters related to the management of the devices connected to the network.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PRIME NB OFDM PLC MAC network administration data		0...n	class_id = 85, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mac_list_multicast_entries	(dyn.)	array				x + 0x08
3. mac_list_switch_table	(dyn.)	array				x + 0x10
4. mac_list_direct_table	(dyn.)	array				x + 0x18
5. mac_list_available_switches	(dyn.)	array				x + 0x20
6. mac_list_phy_comm	(dyn)	array				x + 0x28
Specific methods		***m/o***				
1. reset (data)		*o*				x + 0x30

4.12.9.2 Attribute description

4.12.9.2.1 logical_name

Identifies the “PRIME NB OFDM PLC MAC network administration data” object instance. See 6.2.27.

4.12.9.2.2 mac_list_multicast_entries

PIB attribute 0x0052: List of entries in multicast switching table. This list is not maintained in service nodes in a *Terminal* functional state.

```
mac_list_multicast_entries_type::= array mac_list_multicast_entries_element

mac_list_multicast_entries_element::= structure

{
    mcast_entry_LCID:       integer, -- LCID of multicast group
    mcast_entry_members:    long     -- number of child nodes
}
```

The number of child nodes is the number of the members of this group, including the Node itself.

4.12.9.2.3 mac_list_switch_table

PIB attribute 0x0053: Switch table. This table is not maintained by service nodes in a *Terminal* state.

```
mac switch_table::= array    stbl_entry_LSID

stbl_entry_LSID::=           long   -- SID of attached Switch node
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.9.2.4 mac_list_direct_table

PIB attribute 0x0055: Direct table.

```
mac_direct_table::= array     mac_direct_table_element

mac_direct_table_element::= structure
{
        dconn_entry_src_SID:        long,
        dconn_entry_src_LNID:       long,
        dconn_entry_src_LCID:       long,
        dconn_entry_dst_SID:        long,
        dconn_entry_dst_LNID:       long,
        dconn_entry_dst_LCID:       long,
        dconn_entry_DID:            octet-string (size 6 bytes)
}
```

Where:

– dconn_entry_src_SID is the SID of switch through which the source service node is connected;

– dconn_entry_src_LNID is the NID allocated to the source service node;

– dconn_entry_src_LCID is the LCID allocated to this connection at the source;

– dconn_entry_dst_SID is the SID of the switch through which the destination service node is connected;

– dconn_entry_dst_LNID is the NID allocated to the destination service node;

– dconn_entry_dst_LCID is the LCID allocated to this connection at the destination;

– dconn_entry_DID is the EUI-48 of the direct switch.

4.12.9.2.5 mac_list_available_switches

PIB attribute 0x0056: List of switch nodes whose beacons are received.

```
mac_list_available_switches::= array  mac_list_available_switches_element

mac_list_available_switches_element::= structure

{
        slist_entry_SNA:        octet-string (size 6 bytes),
        slist_entry_LSID:       long,
        slist_entry_level:      integer,
        slist_entry_rx_level:   integer,
        slist_entry_rx_snr:     integer
}
```

Where:

– slist_entry_SNA is EUI-48 of the subnetwork;

– slist_entry_LSID is SID of this switch;

– slist_entry_level is level of this switch in subnetwork hierarchy;

– slist_entry_rx_level is the received signal level for this Switch;

– slist_entry_rx_snr is the signal to noise ratio for this switch.

4.12.9.2.6 mac_list_phy_comm

PIB attribute 0x0057: List of PHY communication parameters. It is maintained in every node. For terminal nodes it contains only one entry for the switch the node is connected through. For other nodes is contains also entries for every directly connected child node.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
mac_list_phy_comm::= array      phy_comm_element

phy_comm_element::= structure

{
phy_Comm_EUI:              octet-string,
phy_Comm_Tx_Pwr:           integer,
phy_Comm_Tx_Cod:           integer,
phy_Comm_Rx_Cod:           integer,
phy_Comm_Rx_Lvl:           integer,
phy_Comm_SNR:              integer,
phy_Comm_Tx_Pwr_Mod:       integer,
phy_Comm_Tx_Cod_Mod:       integer,
phy_Comm_Rx_Cod_Mod:       integer
}
```

Where:

– phy_Comm_EUI is the EUI-48 of the other device;
– phy_Comm_Tx_Pwr is the Tx power of GPDU packets sent to the device;
– phy_Comm_Tx_Cod is the Tx coding of GPDU packets sent to the device;
– phy_Comm_Rx_Cod is the Rx coding of GPDU packets received from the device;
– phy_Comm_Rx_Lvl is the Rx power level of GPDU packets received from the device;
– phy_Comm_SNR is the SNR of GPDU packets received from the device;
– phy_Comm_Tx_Pwr_Mod is the number of times the Tx power was modified;
– phy_Comm_Tx_Cod_Mod is the number of times the Tx coding was modified;
– phy_Comm_Rx_Cod_Mod is the number of times the Rx coding was modified.

4.12.9.3 Method description

4.12.9.3.1 reset (data)

This method is used for resetting all the entries (to an array of 0 elements) of the attributes 2 to 6 of the instance of this interface class.

```
data::= integer (0)
```

4.12.10 PRIME NB OFDM PLC MAC address setup (class_id = 43, version = 0)

An instance of the MAC address setup IC holds the EUI-48 MAC address of the device. The size of this octet string is 6 due to the fact that this address is a EUI-48 and is unique. See also 4.9.4 and 6.2.27.

4.12.11 PRIME NB OFDM PLC Application identification (class_id = 86, version = 0)

4.12.11.1 Overview

An instance of the “PRIME NB OFDM PLC Application identification IC” holds identification information related to administration and maintenance of PRIME NB OFDM PLC devices. They are not communication parameters but allow the device management.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PRIME NB OFDM PLC Application identification		0...n	class_id = 86, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. firmware_version	(static)	octet-string		128		x + 0x08
3. vendor_Id	(static)	long-unsigned				x + 0x10
4. product_Id	(static)	long-unsigned				x + 0x18
Specific methods		***m/o***				

4.12.11.2 Attribute description

4.12.11.2.1 logical_name

Identifies the device setup object instance. See 6.2.27.

4.12.11.2.2 firmware_version

PIB attribute 0x0075: Textual description of the firmware version running on the device.

4.12.11.2.3 vendor_Id

PIB attribute 0x0076: Unique vendor identifier assigned by PRIME Alliance.

4.12.11.2.4 product_Id

PIB attribute 0x0077: Vendor assigned unique identifier for specific product.

4.13 Interface classes for setting up and managing the DLMS®/COSEM narrowband OFDM PLC profile for G3-PLC networks

4.13.1 Overview

This subclause 4.13 specifies version 1 of interface classes for setting up and managing the MAC and 6LoWPAN Adaptation layers of the DLMS®/COSEM G3-PLC profile, based on ITU-T G.9903 Amd. 1:2017. See also Annex E.

NOTE 1 ITU-T G.9903:2013 was supported using the version 0 of these interface classes which are retained in 5.13.

NOTE 2 ITU-T G.9903:2014 was supported using version 1 of the G3-PLC MAC layer counters in section 5.13.5, version 1 of the G3-PLC MAC setup which is retained in 5.13.6, and G3-PLC 6LoWPAN adaptation layer setup version 1 which is retained in 5.13.7.

For this purpose, the elements of the PAN Information Base (PIB) have been mapped to attributes of COSEM ICs.

COSEM objects for data exchange using G3-PLC, if implemented, shall be located in the Management Logical Device of COSEM servers.

To set up and manage the DLMS®/COSEM G3-PLC profile layers (including PHY, IEEE 802.15.4:2006 MAC and 6LoWPAN), three ICs are specified:

- “G3-PLC MAC layer counters”, see 4.13.3;
- “G3-PLC MAC setup”, see 4.13.4;
- “G3-PLC 6LoWPAN adaptation layer setup”, see 4.13.5.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

An instance of the existing COSEM interface class “MAC address” (class_id = 43, version = 0) is needed to indicate the EUI-48 MAC address of the G3-PLC modem (corresponding to aExtendedAddress constant in IEEE 802.15.4:2006).

IPv6 configuration is provided by an instance of “IPv6 setup” class.

NOTE 3 The PHY layer of ITU-T G.9903 Amd. 1:2017 is out of scope of the G3-PLC setup ICs.

4.13.2 Mapping of G3-PLC PIB attributes to COSEM IC attributes

In terms of IEEE 802.15.4:2006, a meter is a Reduced Function Device (RFD) while a concentrator / Neighbourhood Network Access Point (NNAP) is a Full Function Device (FFD) / PAN coordinator. In terms of DLMS®/COSEM the meter is the server and the concentrator / NNAP is the client (or an agent for a client).

As COSEM models only the server and not the client, the G3-PLC setup classes concern only the RFD (Reduced Function Device) and not the PAN coordinator.

Table 41 shows the mapping of G3-PLC PIB attributes to attributes of COSEM interface classes.

Table 41 – Mapping of G3-PLC IB attributes to COSEM IC attributes

Name	Identifier	Interface class	class_id / attribute
MAC counters – Read only PIB attributes that provide statistic information [1]			
mac_Tx_data_packet_count	0x0101	G3-PLC MAC layer counters (class_id = 90, version = 1)	90 / Att. 2
mac_Rx_data_packet_count	0x0102		90 / Att. 3
mac_Tx_cmd_packet_count	0x0103		90 / Att. 4
mac_Rx_cmd_packet_count	0x0104		90 / Att. 5
mac_CSMA_fail_count	0x0105		90 / Att. 6
mac_CSMA_no_ACK_count	0x0106		90 / Att. 7
mac_bad_CRC_count	0x0109		90 / Att. 8
mac_Tx_data_broadcast_count	0x0108		90 / Att. 9
mac_Rx_data_broadcast_count	0x0107		90 / Att. 10
MAC setup PIB attributes – Read only and read-write and write only variables [1,2]			
mac_short_address	0x0053	G3-PLC MAC setup (class_id = 91, version = 2)	91 / Att. 2
mac_RC_coord	0x010F		91 / Att. 3
mac_PAN_id	0x0050		91 / Att. 4
mac_key_table	0x0071		91 / Att. 5
mac_frame_counter	0x0077		91 / Att. 6
mac_tone_mask	0x0110		91 / Att. 7
mac_TMR_TTL	0x010D		91 / Att. 8
mac_max_frame_retries	0x0059		91 / Att. 9
mac_neighbour_table_entry_TTL	0x010E		91 / Att. 10
mac_neighbour_table	0x010A		91 / Att. 11
mac_high_priority_window_size	0x0100		91 / Att. 12
mac_CSMA_fairness_limit	0x010C		91 / Att. 13
mac_beacon_randomization_window_length	0x0111		91 / Att. 14
mac_A	0x0112		91 / Att. 15
mac_K	0x0113		91 / Att. 16

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Name	Identifier	Interface class	class_id / attribute
mac_min_CW_attempts	0x0114		91 / Att. 17
mac_cenelec_legacy_mode	0x0115		91 / Att. 18
mac_FCC_legacy_mode	0x0116		91 / Att. 19
mac_max_BE	0x0047		91 / Att. 20
mac_max_CSMA_backoff	0x004E		91 / Att. 21
mac_min_BE	0x004F		91 / Att. 22
6LoWPAN adaptation layer IB attributes – Read only and read-write variables [3, 4]			
adp_max_hops	0x0F	G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 2)	92 / Att. 2
adp_weak_LQI_value	0x1A		92 / Att. 3
adp_security_level	0x00		92 / Att. 4
adp_prefix_table	0x01		92 / Att. 5
adp_routing_configuration	0x09, 0x0A, 0x0D, 0x11-0x19, 0x1B, 0x1F		92 / Att. 6
adp_broadcast_log_table_entry_TTL	0x02		92 / Att. 7
adp_routing_table	0x0C		92 / Att. 8
adp_context_information_table	0x07		92 / Att. 9
adp_blacklist_table	0x1E		92 / Att. 10
adp_broadcast_log_table	0x0B		92 / Att. 11
adp_group_table	0x0E		92 / Att. 12
adp_max_join_wait_time	0x20		92 / Att. 13
adp_path_discovery_time	0x21		92 / Att. 14
adp_active_key_index	0x22		92 / Att. 15
adp_metric_type	0x03		92 / Att. 16
adp_coord_short_address	0x08		92 / Att. 17
adp_disable_default_routing	0xF0		92 / Att. 18
adp_device_type	0x10		92 / Att. 19

[1] See ITU-T G.9903 Amd. 1:2017, 9.3.6.2.2 and 9.3.6.2.3.

[2] The following attributes of the G3-PLC MAC sublayer IB attributes have been excluded as there is no need to expose them: *macBSN, macDSN, macAckWaitDuration, macFreqNotching, macTimeStampSupported, macPromiscuousMode, macSecurityEnabled*.

[3] See ITU-T G.9903 Amd. 1:2017, 9.4.1.1.

[4] The following attributes of the G3-PLC Adaptation sublayer IB attributes have been excluded as there is no need to expose them; *adpSoftVersion, adpSnifferMode*.

NOTE Whereas in ITU-T G.9903 Amd. 1:2017 the camel-case notation is used, in COSEM interface class specifications – and in this table – the underscore notation is used.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.3 G3-PLC MAC layer counters (class_id = 90, version = 1)

4.13.3.1 Overview

An instance of the “G3-PLC MAC layer counters” IC stores counters related to the MAC layer exchanges. The objective of these counters is to provide statistical information for management purposes.

The attributes of instances of this IC shall be read only. They can be reset using the reset method.

G3-PLC MAC layer counters		0...n	class_id = 90, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mac_Tx_data_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x08
3. mac_Rx_data_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x10
4. mac_Tx_cmd_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x18
5. mac_Rx_cmd_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x20
6. mac_CSMA_fail_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x28
7. mac_CSMA_no_ACK_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x30
8. mac_bad_CRC_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x38
9. mac_Tx_data_broadcast_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x40
10. mac_Rx_data_broadcast_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x48
Specific methods		***m/o***				
1. reset (data)		*O*				x + 0x50

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.3.2 Attribute description

NOTE When a counter reaches the maximum value (0xFFFFFFFF), it's automatically rolled-over.

4.13.3.2.1 logical_name

Identifies the "G3-PLC MAC layer counters" object instance. See 6.2.28.

4.13.3.2.2 mac_Tx_data_packet_count

PIB attribute 0x0101: Statistic counter of successfully transmitted data packets (MSDUs).

4.13.3.2.3 mac_Rx_data_packet_count

PIB attribute 0x0102: Statistic counter of successfully received data packets (MSDUs).

4.13.3.2.4 mac_Tx_cmd_packet_count

PIB attribute 0x0103: Statistic counter of successfully transmitted command packets.

4.13.3.2.5 mac_Rx_cmd_packet_count

PIB attribute 0x0104: Statistic counter of successfully received command packets.

4.13.3.2.6 mac_CSMA_fail_count

PIB attribute 0x0105: Counts the number of times when CSMA backoffs reach macMaxCSMABackoffs.

4.13.3.2.7 mac_CSMA_no_ACK_count

PIB attribute 0x0106: Counts the number of times when an ACK is not received while transmitting a unicast data frame (The loss of ACK is attributed to collisions).

4.13.3.2.8 mac_bad_CRC_count

PIB attribute 0x0109: Statistic counter of the number of frames received with bad CRC.

4.13.3.2.9 mac_Tx_data_broadcast_count

PIB attribute 0x0108: Statistic counter of the number of broadcast frames sent.

4.13.3.2.10 mac_Rx_data_broadcast_count

PIB attribute 0x0107: Statistic counter of successfully received broadcast packets.

4.13.3.3 Method description

4.13.3.3.1 reset (data)

This method forces a reset of the object. By invoking this method, the value of all counters is set to 0.

data::= integer (0)

4.13.4 G3-PLC MAC setup (class_id = 91, version = 2)

4.13.4.1 Overview

An instance of the “G3-PLC MAC setup” IC holds the necessary parameters to set up and manage the G3-PLC IEEE 802.15.4:2006 MAC sub-layer.

These attributes influence the functional behaviour of an implementation. Implementations may allow changes to the attributes during normal running, i.e. even after the device start-up sequence has been executed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3-PLC MAC setup		0...n	class_id = 91, version = 1			
Attributes		**Data type**	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mac_short_address	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x08
3. mac_RC_coord	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x10
4. mac_PAN_id	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x18
5. mac_key_table	(dyn.)	array				x + 0x20
6. mac_frame_counter	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x28
7. mac_tone_mask	(static)	bit-string			0x00000000FFFFFFFF	x + 0x30
8. mac_TMR_TTL	(static)	unsigned	0	255	2	x + 0x38
9. mac_max_frame_retries	(static)	unsigned	0	10	5	x + 0x40
10. mac_neighbour_table_entry_TTL	(static)	unsigned	0	255	255	x + 0x48
11. mac_neighbour_table	(dyn.)	array				x + 0x50
12. mac_high_priority_window_size	(static)	unsigned	1	7	7	x + 0x58
13. mac_CSMA_fairness_limit	(static)	unsigned	See below	255	25	x + 0x60
14. mac_beacon_randomization_window_length	(static)	unsigned	1	254	12	x + 0x68
15. mac_A	(static)	unsigned	3	20	8	x + 0x70
16. mac_K	(static)	unsigned	1	See below	5	x + 0x78
17. mac_min_CW_attempts	(static)	unsigned	0	255	10	x + 0x80
18. mac_cenelec_legacy_mode	(static)	unsigned	0	255	1	x + 0x88
19. mac_FCC_legacy_mode	(static)	unsigned	0	255	1	x + 0x90
20. mac_max_BE	(static)	unsigned	0	20	8	x + 0x98
21. mac_max_CSMA_backoff	(static)	unsigned	0	255	50	x + 0xA0
22. mac_min_BE	(static)	unsigned	0	20	3	x + 0xA8
Specific methods		***m/o***				
1. mac_get_neighbour_table_entry (data)		o				x + 0xB0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2 Attribute description

4.13.4.2.1 logical_name

Identifies the “G3-PLC MAC setup” object instance. See 6.2.28.

4.13.4.2.2 mac_short_address

PIB attribute 0x0053: The 16-bit address the device is using to communicate through the PAN. Its value shall be equal to 0xFFFF when the device does not have a short address. An associated device necessarily has a short address, so that a device cannot be in the state where it is associated but does not have a short address.

4.13.4.2.3 mac_RC_coord

PIB attribute 0x010F: Route cost to coordinator, to be used in the beacon payload as RC_COORD.

4.13.4.2.4 mac_PAN_id

PIB attribute 0x0050: The 16-bit identifier of the PAN through which the device is operating. A value equal to 0xFFFF indicates that the device is not associated.

4.13.4.2.5 mac_key_table

PIB aattribute 0x0071: This attribute holds GMK keys required for MAC layer ciphering. The attribute can hold up to two 16-bytes keys. The Key Identifier value shall be different for each key.

For security reason, the key entries cannot be read, only written.

```
array mac_GMK

mac_GMK::= structure

{
    key_id:    unsigned,
    key:  octet-string
}
```

Where:

key_id The Key Identifier used to refer to this key, can take the value 0 or 1.

key The AES-128 key used for ciphering the frames exchanged at MAC layer.

4.13.4.2.6 mac_frame_counter

PIB attribute 0x0077: The outgoing frame counter for this device, used when ciphering frames at MAC layer.

4.13.4.2.7 mac_tone_mask

PIB attribute 0x0110: Defines the tone mask to use during symbol formation.

4.13.4.2.8 mac_TMR_TTL

PIB attribute 0x010D: Maximum time to live of tone map parameters entry in the neighbour table in minutes.

NOTE This attribute was used with different definitions in earlier versions of the Interface Class. See 5.13.6.2.8.

4.13.4.2.9 mac_max_frame_retries

PIB attribute 0x0059: Maximum number of retransmissions.

4.13.4.2.10 mac_neighbour_table_entry_TTL

PIB attribute 0x010E: Maximum time to live for an entry in the neighbour table in minutes.

NOTE This attribute was used with different name and different definitions in earlier versions of the Interface Class. See 5.13.6.2.10.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2.11 mac_neighbour_table

PIB attribute 0x010A: See ITU-T G.9903:2017 ,9.3.7.2 for CENELEC and FCC bands.

The neighbour table contains information about all the devices within the POS of the device. One element of the table represents one PLC direct neighbour of the device.

```
array neighbour_table

neighbour_table::= structure

{
    short_address:                  long-unsigned,
    payload_modulation_scheme:      boolean,
    tone_map:                       bit-string,
    modulation_type:                    enum,
    tx_gain:                        integer,
    tx_res:                         enum,
    tx_coeff:                       bit-string,
    reverse_lqi:                        unsigned,
    phase_differential              integer,
    TMR_valid_time:                 unsigned,
    no_data:                        unsigned
}
```

Where:

short_address — The MAC Short Address of the node which this entry refers to.

payload_modulation_scheme — Payload Modulation scheme to be used when transmitting to this neighbour.
FALSE: Differential,
TRUE: Coherent

tone_map — The Tone Map parameter defines which frequency sub-band can be used for communication with the device. A bit set to 1 means that the frequency sub-band can be used, and a bit set to 0 means that frequency sub-band shall not be used.

modulation — The modulation type to use for communicating with the device.

enum:

(0) Robust Mode,
(1) DBPSK,
(2) DQPSK,
(3) D8PSK,
(4) 16-QAM

NOTE 1 The 16-QAM modulation is optional and only applicable for FCC band.

tx_gain — Defines the Tx Gain to use to transmit frames to that device.

tx_res — Defines the Tx Gain resolution corresponding to one gain step.
0: 6 dB,
1: 3 dB

tx_coeff — A parameter that specifies transmitter gain for each group of tones represented by one valid bit of the tone map.

NOTE 2 One group of tones gathers 6 consecutive tones (or carriers) or CENELEC bands, and 3 consecutive tones for FCC band.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The receiver measures the frequency-dependent attenuation of the channel and may request the transmitter to compensate for this attenuation by increasing the transmit power on sections of the spectrum that are experiencing attenuation in order to equalize the received signal. Each group of tones is mapped to a 4-bit value for CENELEC-A or a 2-bit value for FCC where a "0" in the most significant bit indicates a positive gain value, hence an increase in the transmitter gain scaled by TXRES is requested for that section and a "1" indicates a negative gain value, hence a decrease in the transmitter gain scaled by TXRES is requested for that section. Implementing this feature is optional and it is intended for frequency selective channels. If this feature is not implemented, the value zero shall be used.

In the CENELEC bandplan, the first bit in the bit-string (bit 0) corresponds to TXCOEF[0] and the last bit in the bit-string (bit 23) corresponds to TXCOEF[23], noting that the bits are ordered in groups as defined in ITU-T G.9903 Amd. 1:2017, Table 9-21.

In the FCC bandplan, the first bit in the bit-string (bit 0) corresponds to TXCOEF[0] and the last bit in the bit-string (bit 47) corresponds to TXCOEF[47], noting that the bits are ordered in groups as defined in ITU-T G.9903 Amd. 1:2017, Table 9-22.

NOTE 3 A description of tx_coeff is included in G.9903:2017, Tables 9-21 and 9-22.

In the CENELEC-A bandplan, each group of 4 bits of the TXCOEF field is associated with one sub-band consisting of a group of 6 tones.

Bit number	TXCOEF field	Sub-band (kHz)
0	TXCOEF[0]	35,937 5 to 43,75
1	TXCOEF[1]	
2	TXCOEF [2]	
3	TXCOEF [3]	
4	TXCOEF [4]	45,312 5 to 53,125
5	TXCOEF [5]	
6	TXCOEF[6]	
7	TXCOEF[7]	
8	TXCOEF[8]	54,6875 to 62,5
9	TXCOEF[9]	
10	TXCOEF[10]	
11	TXCOEF[11]	
12	TXCOEF[12]	64,062 5 to 71,875
13	TXCOEF[13]	
14	TXCOEF[14]	
15	TXCOEF[15]	
16	TXCOEF[16]	73,437 5 to 81,25
17	TXCOEF[17]	
18	TXCOEF[18]	
19	TXCOEF[19]	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Bit number	TXCOEF field	Sub-band (kHz)
20	TXCOEF[20]	82,812 5 to 90,625
21	TXCOEF[21]	
22	TXCOEF[22]	
23	TXCOEF[23]	

In the FCC bandplan, each group of 2 bits of the TXCOEF field is associated with one sub-band consisting of a group of 3 tones

Bit number	TXCOEF field	Sub-band (kHz)
0	TXCOEF[0]	154,687 5 to 164,062 5
1	TXCOEF[1]	
2	TXCOEF [2]	168,75 to 178,125
3	TXCOEF [3]	
4	TXCOEF [4]	182,812 5 to 192,187 5
5	TXCOEF [5]	
6	TXCOEF[6]	196,875 to 206,25
7	TXCOEF[7]	
8	TXCOEF[8]	210,937 5 to 220,312 5
9	TXCOEF[9]	
10	TXCOEF[10]	225 to 234,375
11	TXCOEF[11]	
12	TXCOEF[12]	239,062 5 to 248,437 5
13	TXCOEF[13]	
14	TXCOEF[14]	253,125 to 262,5
15	TXCOEF[15]	
16	TXCOEF[16]	267,187 5 to 276,562 5
17	TXCOEF[17]	
18	TXCOEF[18]	281,25 to 290,625
19	TXCOEF[19]	
20	TXCOEF[20]	295,312 5 to 304,687 5
21	TXCOEF[21]	
22	TXCOEF[22]	309,375 to 318,75
23	TXCOEF[23]	
24	TXCOEF[24]	323,437 5 to 332,812 5
25	TXCOEF[25]	
26	TXCOEF[26]	337,5 to 346,875
27	TXCOEF[27]	
28	TXCOEF[28]	351,562 5 to 360,937 5
29	TXCOEF[29]	
30	TXCOEF[30]	365,625 to 375
31	TXCOEF[31]	
32	TXCOEF[32]	379,687 5 to 389,062 5
33	TXCOEF[33]	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Bit number	TXCOEF field	Sub-band (kHz)
34	TXCOEF[34]	393,75 to 403,125
35	TXCOEF[35]	
36	TXCOEF[36]	407,812 5 to 417,187 5
37	TXCOEF[37]	
38	TXCOEF[38]	421,875 to 431,25
39	TXCOEF[39]	
40	TXCOEF[40]	435,937 5 to 445,312 5
41	TXCOEF[41]	
42	TXCOEF[42]	450 to 459,375
43	TXCOEF[43]	
44	TXCOEF[44]	464,062 5 to 473,437 5
45	TXCOEF[45]	
46	TXCOEF[46]	478,125 to 487,5
47	TXCOEF[47]	

reverse_lqi — Link Quality Indicator of the link to the neighbour (reverse LQI)

NOTE 4 LQI value measured during reception of the PPDU from the neighbour. The LQI measurement is a characterization of the strength and/or quality of a received packet.

phase_differential — Phase difference in multiples of 60 degrees between the mains phase of the local node and the neighbour node. PhaseDifferential can assume six integer values between 0 and 5.

TMR_valid_time — Remaining time in minutes until which the tone map response parameters in the neighbour table are considered valid.

– When the entry is created, this value shall be set to the default value 0.

– When it reaches 0, a tone map request may be issued if data is sent to this device. Upon successful reception of a tone map response, this value is set to mac_TMR_TTL.

no_data — Value that may be used as needed for some implementations as determined by a companion specification

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2.12 mac_high_priority_window_size

PIB attribute 0x0100: The high priority contention window size in number of slots.

4.13.4.2.13 mac_CSMA_fairness_limit

PIB attribute 0x010C: Channel access fairness limit. Specifies how many failed back-off attempts, back-off exponent is set to minBE. This attribute can take a value between 2 × (macMaxBE – macMinBE) and 255.

4.13.4.2.14 mac_beacon_randomization_window_length

PIB attribute 0x0111: Duration time in seconds for the beacon randomization.

4.13.4.2.15 mac_A

PIB attribute 0x0112: This parameter controls the adaptive CW linear decrease.

4.13.4.2.16 mac_K

PIB attribute 0x0113: Rate adaptation factor for channel access fairness limit. This attribute can take a value between 1 and macCSMAFairnessLimit.

4.13.4.2.17 mac_min_CW_attempts

PIB attribute 0x0114: Number of consecutive attempts while using minimum CW.

4.13.4.2.18 mac_cenelec_legacy_mode

PIB attribute 0x0115: This read only attribute indicates the capability of the node.

0: The following configuration is used (legacy mode):

- Elementary interleaving;
- Interleaver parameters n_i and n_j are not swapped when I(i,j) = 0.

1: The following configuration is used (non legacy mode):

- Full Block interleaving;
- Interleaver parameters n_i and n_j are swapped when I(i,j) = 0.

4.13.4.2.19 mac_FCC_legacy_mode

PIB attribute 0x0116: This read only attribute indicates the capability of the node.

0: The following configuration is used (legacy mode):

- Differential FCH modulation;
- Elementary interleaving;
- Interleaver parameters n_i and n_j are not swapped when I(i,j) = 0;
- Single RS block.

1: The following configuration is used (non legacy mode):

- Coherent FCH modulation;
- Full Block interleaving;
- Interleaver parameters n_i and n_j are swapped when I(i,j) = 0;
- Two RS blocks.

4.13.4.2.20 mac_max_BE

PIB attribute 0x0047: Maximum value of backoff exponent. It should always be greater than macMinBE.

4.13.4.2.21 mac_max_CSMA_backoff

PIB attribute 0x004E: Maximum number of backoff attempts.

4.13.4.2.22 mac_min_BE

PIB attribute 0x004F: Minimum value of backoff exponent.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.3 Method description

4.13.4.3.1 mac_get_neighbour_table_entry (data)

This method is used to retrieve the mac neighbour table for one MAC short address. It may be used to perform topology monitoring by the client.

The method invocation parameter contains a mac_short_address.

data::= long-unsigned

The response parameter includes the neighbour table for this mac_short_address.

data::= array neighbour_table

where mac_neighbour_table is as defined in the *mac_neighbour_table* attribute of the present IC.

4.13.5 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 2)

4.13.5.1 Overview

An instance of the “G3-PLC 6LoWPAN adaptation layer setup” IC holds the necessary parameters to set up and manage the G3-PLC 6LoWPAN Adaptation layer.

These attributes influence the functional behaviour of an implementation. Implementations may allow changes to their values during normal running, i.e. even after the device start-up sequence has been executed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3-PLC 6LoWPAN adaptation layer setup		0...n	class_id = 92, version = 2			
Attribute (s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. adp_max_hops	(static)	unsigned	1	14	8	x + 0x08
3. adp_weak_LQI_value	(static)	unsigned	0	255	52	x + 0x10
4. adp_security_level	(static)	unsigned	0	5	5	x + 0x18
5. adp_prefix_table	(dyn)	array				x + 0x20
6. adp_routing_configuration	(static)	array				x + 0x28
7. adp_broadcast_log_table_entry_TTL	(static)	long- unsigned	0	65535	2	x + 0x30
8. adp_routing_table	(dyn)	array				x + 0x38
9. adp_context_information_table	(dyn)	array				x + 0x40
10. adp_blacklist_table	(dyn)	array				x + 0x48
11. adp_broadcast_log_table	(dyn)	array				x + 0x50
12. adp_group_table	(dyn)	array				x + 0x58
13. adp_max_join_wait_time	(static)	long- unsigned	0	1023	20	x + 0x60
14. adp_path_discovery_time	(static)	unsigned	0	255	40	x + 0x68
15. adp_active_key_index	(static)	unsigned	0	1	0	x + 0x70
16. adp_metric_type	(static)	unsigned	0x00	0x0F	0x0F	x + 0x78
17. adp_coord_short_address	(static)	long- unsigned	0x0000	0x7FFF	0x0000	x + 0x80
18. adp_disable_default_routing	(static)	boolean			FALSE	x + 0x88
19. adp_device_type	(static)	enum	0	2	2	x + 0x90
20. adp_default_coord_route_enabled	(static)	boolean			FALSE	x + 0x98
21. adp_destination_address_set	(dyn)	array				x + 0xA0
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.5.2 Attribute description

4.13.5.2.1 logical_name

Identifies the “G3-PLC OFDM 6LoWPAN adaptation layer setup” object instance. See 6.2.28.

4.13.5.2.2 adp_max_hops

PIB attribute 0x0F: Defines the maximum number of hops to be used by the routing algorithm.

4.13.5.2.3 adp_weak_LQI_value

PIB attribute 0x1A: The weak link value defines the LQI value below which a link to a neighbour is considered as a weak link. A value of 52 represents an SNR of 3 dB.

4.13.5.2.4 adp_security_level

PIB attribute 0x00: The minimum security level to be used for incoming and outgoing adaptation frames. Only values 0 (no ciphering) and 5 (ciphering with 32 bits integrity code) are supported.

4.13.5.2.5 adp_prefix_table

PIB attribute 0x01: Contains the list of prefixes defined on this PAN.

NOTE It is assumed that the link local IPv6 address exists independently and is not affected by the prefixes defined in the prefix table.

4.13.5.2.6 adp_routing_configuration

The routing configuration element specifies all parameters linked to the routing mechanism described in ITU-T G.9903 Amd. 1:2017. The elements are specified in 9.4.1.2 of that Recommendation.

NOTE 1 The Link cost calculation is provided in ITU-T G.9903:2017, Annex B.

```
array routing_configuration
routing_configuration::= structure
{
        adp_net_traversal_time:          unsigned,
        adp_routing_table_entry_TTL:     long-unsigned,
        adp_Kr:                          unsigned,
        adp_Km:                          unsigned,
        adp_Kc:                          unsigned,
        adp_Kq:                          unsigned,
        adp_Kh:                          unsigned,
        adp_Krt:                         unsigned,
        adp_RREQ_retries:                unsigned,
        adp_RREQ_wait:                   unsigned,
        adp_blacklist_table_entry_TTL:   long-unsigned,
        adp_unicast_RREQ_gen_enable:     boolean,
        adp_RLC_Enabled:                 boolean,
        adp_add_rev_link_cost:           unsigned
}
```

Where

adp_net_traversal_time — PIB attribute 0x11: Maximum time that a packet is expected to take to reach any node from any node in seconds.
Range : 0-255
Default value : 20

adp_routing_table_entry_TTL — PIB attribute 0x12: Maximum time-to-live of a routing table entry (in minutes).
Range : 0-65 535
Default value : 60

adp_Kr — PIB attribute 0x13: A weight factor for the Robust Mode to calculate link cost.
Range : 0-31
Default value : 0

adp_Km — PIB attribute 0x14: A weight factor for modulation to calculate link cost.
Range : 0-31
Default value : 0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

adp_Kc	PIB attribute 0x15: A weight factor for number of active tones to calculate link cost. Range : 0-31 Default value : 0
adp_Kq	PIB attribute 0x16: A weight factor for LQI to calculate route cost. Range : 0-50 Default value : 10 for CENELEC A band / 40 for FCC band.
adp_Kh	PIB attribute 0x17: A weight factor for hop to calculate link cost. Range : 0-31 Default value : 4 for CENELEC A band / 2 for FCC band.
adp_Krt	PIB attribute 0x1B: A weight factor for the number of active routes in the routing table to calculate link cost. Range : 0-31 Default value : 0
adp_RREQ_retries	PIB attribute 0x18: The number of RREQ re-transmission in case of RREP reception time out. Range : 0-255 Default value : 0
adp_RREQ_wait	PIB attribute 0x19: The number of seconds to wait between two consecutive RREQ – RERR generations. Range : 0-255 Default value : 30 NOTE 2 The title and definition of this item has been modified from the previous version.
adp_blacklist_table_entry_TTL	PIB attribute 0x1F: Maximum time-to-live of a blacklisted neighbour entry (in minutes). Range : 0-65 535 Default value : 10
adp_unicast_RREQ_gen_enable	PIB attribute 0x0D: If TRUE, the RREQ shall be generated with its "unicast RREQ" flag set to '1'. If FALSE, the RREQ shall be generated with its "unicast RREQ" flag set to '0'. Default value : TRUE
adp_RLC_enabled	PIB attribute 0x09: Enable the sending of RLCREQ frame by the device. Default value : FALSE
adp_add_rev_link_cost	PIB attribute 0x0A: It represents an additional cost to take into account a possible asymmetry in the link. Range : 0-255 Default value : 0

4.13.5.2.7 adp_broadcast_log_table_entry_TTL

PIB attribute 0x02: Maximum time to live of an adpBroadcastLogTable entry (in minutes).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.5.2.8 adp_routing_table

PIB attribute 0x0C: Contains the routing table.

```
array routing_table

routing_table::= structure

{
    destination_address:    long-unsigned,
    next_hop_address:       long-unsigned,
    route_cost:             long-unsigned,
    hop_count:              unsigned,
    weak_link_count:        unsigned,
    valid_time:             long-unsigned
}
```

NOTE 1 This table is actualized each time a route is built or updated (triggered by data traffic) and each time the TTL timer expires.

Where:

destination_address — Address of the destination.

next_hop_address — Address of the next hop on the route towards the destination.

route_cost — Cumulative link cost along the route towards the destination.

hop_count — Number of hops of the selected route to the destination.
Range: 0-14

NOTE 2 Practically the maximum allowed value is limited by adp_max_hops.

weak_link_count — Number of weak links to destination.
Range: 0-14

NOTE 3 Practically the maximum allowed value is limited by adp_max_hops.

valid_time — Remaining time in minutes until when this entry in the routing table is considered valid.

4.13.5.2.9 adp_context_information_table

PIB attribute 0x07: Contains the context information associated to each CID extension field. See ITU-T G.9903:2014. The elements are specified in Table 9-30 of that Recommendation.

```
array context_information_table

context_information_table::= structure

{
    CID:                    bit-string,
    context_length:         unsigned,
    context:                octet-string,
    C:                      boolean,
    valid_lifetime:         long-unsigned
}
```

Where:

CID — Corresponds to the 4-bit context information used for source and destination addresses (SCI, DCI).
The first bit in the bit-string (bit 0) corresponds to the least significant bit of CID in G.9903:2014, Table 9-30.
Range: 0x00-0x0F

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

context_length	Indicates the length of the carried context (up to 128-bit contexts may be carried). Range: 0-128
context	Corresponds to the carried context used for compression/decompression purposes.
C	Indicates if the context is valid for use in compression. FALSE: Only decompression is allowed, TRUE: Compression and decompression are allowed A context may be used for decompression purposes only. Moreover, recommendations made in RFC 6775 should be followed to take into account the propagation of the context to all nodes of the PAN.
valid_lifetime	Remaining time in minutes during which the context information table is considered valid. It is updated upon reception of the advertised context. Range: 0-65 535

4.13.5.2.10 adp_blacklist_table

PIB attribute 0x1E: Contains the list of the blacklisted neighbours.

```
array blacklisted_neighbour_set

blacklisted_neighbour_set::= structure

{
    blacklisted_neighbour_address:   long-unsigned,
    valid_time:                      long-unsigned
}
```

Where:

blacklisted_neighbour_address	The 16-bit address of the blacklisted neighbour.
valid_time	Remaining time in minutes until which this entry in the blacklisted neighbour table is considered valid.

4.13.5.2.11 adp_broadcast_log_table

PIB attribute 0x0B: Contains the broadcast log table.

NOTE This table provides a list of the broadcast packets recently received by this device.

```
array broadcast_log_table

broadcast_log_table::= structure

{
    source_address:      long-unsigned,
    sequence_number:           unsigned,
    valid_time:          long-unsigned
}
```

Where:

source_address	The 16-bit source address of a broadcast packet. This is the address of the broadcast initiator.
sequence_number	The sequence number contained in the BC0 header.
valid_time	Remaining time in minutes until when this entry in the broadcast log table is considered valid.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.5.2.12 adp_group_table

PIB attribute 0x0E: Contains the group addresses to which the device belongs.

array group_address

group_address::= long-unsigned

group_address Group address to which this node has been subscribed.

4.13.5.2.13 adp_max_join_wait_time

PIB attribute 0x20: Network join timeout in seconds for LBD.

4.13.5.2.14 adp_path_discovery_time

PIB attribute 0x21: Timeout for path discovery in seconds.

4.13.5.2.15 adp_active_key_index

PIB attribute 0x22: Index of the active GMK to be used for data transmission.

4.13.5.2.16 adp_metric_type

PIB attribute 0x03: Metric Type to be used for routing purposes.

4.13.5.2.17 adp_coord_short_address

PIB attribute 0x08: Defines the short address of the coordinator.

4.13.5.2.18 adp_disable_default_routing

PIB attribute 0xF0: If TRUE, the default routing (LOADng) is disabled. If FALSE, the default routing (LOADng) is enabled.

4.13.5.2.19 adp_device_type

PIB attribute 0x10: Defines the type of the device connected to the modem:

enum:

(0) PAN device,
(1) PAN coordinator,
(2) Not Defined

4.13.5.2.20 adp_default_coord_route_enabled

PIB attribute 0x24: If TRUE, the adaptation layer adds a default route to the coordinator after successful completion of the bootstrapping procedure. If FALSE no default route will be created.

4.13.5.2.21 adp_destination_address_set

PIB attribute 0x23: Contains the list of the addresses of the devices for which this LOADng router is providing connectivity.

array D_address
D_address::= long-unsigned

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

D_Address The 16-bit address of a destination node attached to this LOADng router and for which this LOADng router provides connectivity.

4.14 Interface classes for setting up and managing DLMS®/COSEM HS-PLC ISO/IEC 12139-1 neighbourhood networks

4.14.1 Overview

COSEM objects for data exchange using DLMS®/COSEM HS-PLC ISO/IEC 12139-1 neighbourhood networks, if implemented, shall be located in the Management Logical Device of COSEM servers.

For setting up and managing DLMS®/COSEM HS-PLC ISO/IEC 12139-1 neighbourhood networks the following ICs are specified:

- HS-PLC ISO/IEC 12139-1 MAC setup, see 4.14.2;
- HS-PLC ISO/IEC 12139-1 CPAS setup, see 4.14.3;
- HS-PLC ISO/IEC 12139-1 IP SSAS setup, see 4.14.4;
- HS-PLC ISO/IEC 12139-1 HDLC SSAS setup, see 4.14.5.

4.14.2 HS-PLC ISO/IEC 12139-1 MAC setup (class_id = 140, version = 0)

4.14.2.1 Overview

Instances of the "HS-PLC ISO/IEC 12139-1 MAC setup" IC hold parameters necessary to set up and manage the MAC layer of the HS-PLC ISO/IEC 12139-1 profile.

HS-PLC ISO/IEC 12139-1 MAC setup		**0…n**	**class_id = 140 version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name		octet-string				x
2. group_id	(static)	long64-unsigned	0	2^{46}		x + 0x08
3. secondary_group_id	(static)	long64-unsigned	0	2^{46}		x + 0x10
4. station_id	(static)	long64-unsigned	0	2^{48}		x + 0x18
5. parent_ station_id	(static)	long64-unsigned	0	2^{48}		x + 0x20
6. repeater_status	(static)	boolean				x + 0x28
7. encryption_mode	(static)	enum	1	2	1	x + 0x30
8. initial_encryption_key	(static)	octet-string				x + 0x38
9. rts/cts	(static)	boolean	0	1	0	x + 0x40
Specific methods		***m/o***				

4.14.2.2 Attribute description

4.14.2.2.1 logical_name

Identifies the “HS-PLC ISO/IEC 12139-1 MAC setup” object instance. See 6.2.31.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.14.2.2.2 group_id

Holds the group identifier of HS-PLC ISO/IEC 12139-1. Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

4.14.2.2.3 secondary_group_id

Holds the secondary group identifier of HS-PLC ISO/IEC 12139-1. Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

4.14.2.2.4 station_id

Holds the station identifier of HS-PLC ISO/IEC 12139-1. Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

4.14.2.2.5 parent_station_id

Holds the parent station identifier* of HS-PLC ISO/IEC 12139-1. Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

* The parent station means the directly neighbouring station from a station itself in a multi-stage HS-PLC ISO/IEC 12139-1 link from the station to NNAP (a source station – Repeater(s) –NNAP).

4.14.2.2.6 repeater_status

Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

Holds the current repeater status of the device.

boolean:

FALSE = no repeater,
TRUE = repeater

4.14.2.2.7 encryption_mode

Holds the encryption mode the PLC station uses. Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

enum:

(0) AES128,
(1) Reserved.

4.14.2.2.8 initial_encryption_key

Encryption key which is used initially during PLC registration (cell join) period. Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

4.14.2.2.9 rts/cts

Holds the status of the RTS/CTS parameter. The RTS/CTS (Request To Send / Clear To Send) parameter is used to prevent collision caused by hidden HS-PLC ISO/IEC 12139-1 stations). Refer to ISO/IEC 12139-1:2009, Clause 7 for detailed information.

RTS/CTS enable/disable

boolean:
FALSE = Disable,
TRUE = Enable

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.14.3 HS-PLC ISO/IEC 12139-1 CPAS setup (class_id = 141, version = 0)

4.14.3.1 Overview

Instances of the "HS-PLC ISO/IEC 12139-1 CPAS setup" IC hold parameters necessary to set up and manage the CPAS layer of the HS-PLC ISO/IEC 12139-1 profile.

HS-PLC ISO/IEC 12139-1 CPAS setup		**0...n**	**class_id = 141 version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name		octet-string				x
2. cpas_address	(static)	long64-unsigned	0	2^{48}		x + 0x08
3. cpas_ether_type	(static)	long-unsigned				x + 0x10
4. master_station_cpas_address	(static)	long64-unsigned	0	2^{48}		x + 0x18
Specific methods		**m/o**				

4.14.3.2 Attribute description

4.14.3.2.1 logical_name

Identifies the "HS-PLC ISO/IEC 12139-1 CPAS setup" object instance. See 6.2.31.

4.14.3.2.2 cpas_address

Holds the CPAS address of the HS-PLC ISO/IEC 12139-1 profile.

See IEC 62056-8-6:2017, 5.4.2.

4.14.3.2.3 cpas_ether_type

Holds the EtherType value of the CPAS sublayer.

See IEC 62056-8-6:2017, 5.4.2.

4.14.3.2.4 master_station_cpas_address

Holds the master station's* CPAS address of the HS-PLC ISO/IEC 12139-1 profile.

* The master station means the destination station of the CPAS frame (i.e. typically NNAP) in a multi-stage HS-PLC ISO/IEC 12139-1 link (a source station – Repeater(s) – A destination station).

4.14.4 HS-PLC ISO/IEC 12139-1 IP SSAS setup (class_id = 142, version = 0)

4.14.4.1 Overview

Instances of the "HS-PLC ISO/IEC 12139-1IP SSAS setup" IC hold parameters necessary to set up and manage the IP SSAS of the HS-PLC ISO/IEC 12139-1 profile.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

HS-PLC ISO/IEC 12139-1 IP SSAS setup	0...n	class_id = 142, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	octet-string				x
2. ip_header_comp_type (static)	enum				x + 0x08
3. ip_alive_time (static)	long-unsigned		0xFFFF	0	x + 0x10
Specific methods	**m/o**				

4.14.4.2 Attribute description

4.14.4.2.1 logical_name

Identifies the "HS-PLC ISO/IEC 12139-1 IP SSAS setup" object instance. See 6.2.31.

4.14.4.2.2 ip_header_comp_type

Holds the IP_Header_Comp_Type value as specified in IEC 62056-8-6:2017, Table 3.

enum:

(0) General IPv4 packet (No compression),
(1) General IPv6 packet (No compression),
(2) Van Jacobson header compression (RFC 1144),
(3) IP header compression (RFC 2508),
(4) ROHC (RFC 3095).

4.14.4.2.3 ip_alive_time

Holds the IP SSAS alive time value in seconds.

0: No expiry

4.14.5 HS-PLC ISO/IEC 12139-1 HDLC SSAS setup (class_id = 143, version = 0)

4.14.5.1 Overview

Instances of the "HS-PLC ISO/IEC 12139-1 HDLC SSAS setup" IC hold parameters necessary to set up and manage the HDLC SSAS of the HS-PLC ISO/IEC 12139-1 profile.

HS-PLC ISO/IEC 12139-1 HDLC SSAS setup	0...n	class_id = 143, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	octet-string				x
2. master_station_id (static)	long64-unsigned	0	2^{48}		x + 0x08
Specific methods	***m/o***				

4.14.5.2 Attribute description

4.14.5.2.1 logical_name

Identifies the “HS-PLC ISO/IEC 12139-1 HDLC SSAS setup” object instance. See 6.2.31.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.14.5.2.2 master_station_id

Holds the master station identifier of the HS-PLC ISO/IEC 12139-1 network.

4.15 ZigBee® setup classes

4.15.1 Overview

This subclause specifies COSEM interface classes required for the external configuration and management of a ZigBee® network to allow interfacing with a multi-part installation that internally uses ZigBee® communications. ZigBee® is a low-power radio communications technology and open standard that is operated by the ZigBee® Alliance, see www.zigbee.org.

ZigBee® is a registered trademark of the ZigBee® Alliance.

NOTE 1 A multi-part installation is one where the meter provides information and/or services to the householder on behalf of the utility. For example, the meter interacts with an in home display, and/or an external load control switch, and/or a smart appliance, to inform the customer of their usage in real time, to control heating devices, and possibly to disconnect peak loads when supply is constrained. While it is possible that the consumer will control the ZigBee® network, in normal operations the utility will control the radio system. This is to ensure that security is maintained for PAN, so that ZigBee® devices such as load switches controlled by the utility operate in a secure manner.

ZigBee® defines a local network of devices linked by radio, with routing and forwarding of messages and with encryption for privacy at network-level.

NOTE 2 Such a local network is known as a PAN – a Personal Area Network. This name is used in the ZigBee® community as ZigBee® is underpinned by the IEEE 802.15.4:2006 standard, which uses the term PAN. This is broadly equivalent to a HAN (Home Area Network – name used in the context of smart metering in the UK) or PAN.

Each PAN has one device designated as ZigBee® coordinator, which has responsibility for creating and managing the network, and which normally acts also as a ZigBee® Trust Center for the management of ZigBee® network, PCLK's and APS Link keys.

There is a process of PAN creation (and corresponding destruction) which is performed by the coordinator; this declares the existence of the network without any devices apart from the coordinator forming part of it. Other ZigBee® devices can join a network created with cooperation of the coordinator, and equally can choose to leave, or can be invited to leave by the coordinator (this is not currently enforceable). Normally devices are members of the network indefinitely; they do not repeatedly join and leave. To create a PAN the coordinator has to receive an external trigger and needs to have setup information including:

- extended PAN ID;
- link keys or install code (for initial communication with new devices);
- radio channel information.

During the process of creating the PAN, the coordinator scans for nearby radio devices, "exchanges" keys, chooses the short addresses, and confirms use of radio channels. Details of the information available from the ZigBee® servers on each device are also exchanged.

NOTE 3 Full details of the joining process are documented in ZigBee® 053474, the ZigBee® specification. More information on ZigBee® technology can be sought at http://www.zigbee.org/.

Figure 29 shows an example architecture with a Comms hub on the left, that comprises a DLMS®/COSEM server as well as the ZigBee® coordinator. The DLMS®/COSEM server has interface objects needed to set up and control the ZigBee® network and may have other COSEM objects to support a metering application. Further, there are two native ZigBee® devices and another DLMS®/COSEM server – an electricity meter or another meter type – on the right which is also a "normal" ZigBee® network device.

Any ZigBee® device can be "joined" to the network by remote control.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

It is assumed that the Comms hub will also have a further network connection to the WAN, and it is assumed that this is managed by existing DLMS® structures – e.g. PSTN, GSM/3G, PLC, etc. – but this is out of scope of this specification.

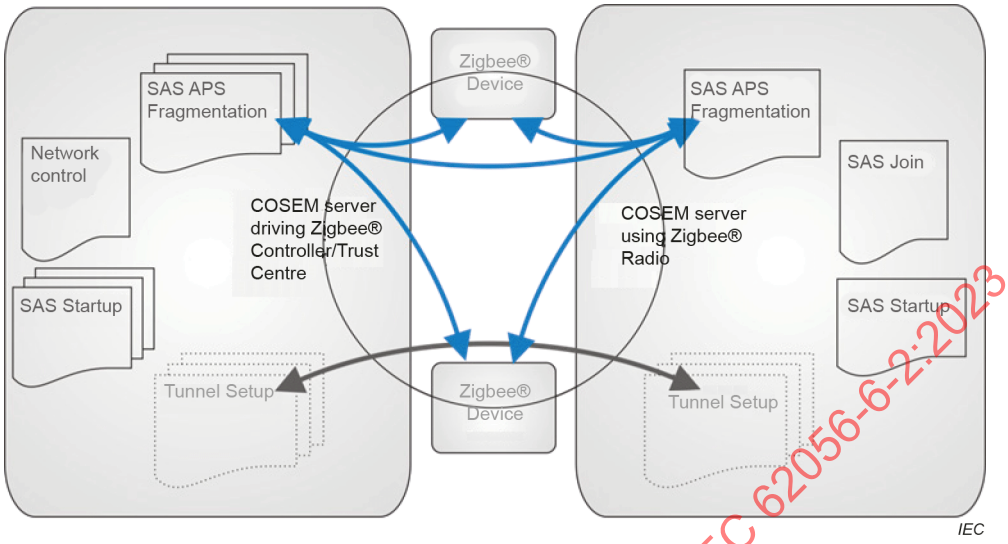


Figure 29 – Example of a ZigBee® network

The role of COSEM interface objects in the process of creating / destructing the PAN is to set up ZigBee® parameters and allow a DLMS®/COSEM client to trigger actions, typically when commissioning the installation, in a system where WAN communications between a central system and a smart meter installation is by means of DLMS®.

Operation of the ZigBee® network is not the responsibility of DLMS®/COSEM. The DLMS®/COSEM server is merely the vehicle for controlling the ZigBee® network by an external manager (DLMS®/COSEM Client).

The use of ZigBee® setup classes in the ZigBee® coordinator and other DLMS®/COSEM ZigBee® devices is shown in Table 42.

Table 42 – Use of ZigBee® setup COSEM interface classes

ZigBee® coordinator	Other DLMS®/COSEM ZigBee® devices	Reference
ZigBee® SAS startup	ZigBee® SAS startup	4.15.2
–	ZigBee® SAS join	4.15.3
ZigBee® SAS APS fragmentation	ZigBee® SAS APS fragmentation	4.15.4
ZigBee® network control	–	4.15.5
Optionally: ZigBee® tunnel setup	Optionally: ZigBee® tunnel setup	4.15.6

This set of COSEM ICs supports the ZigBee® 2007 and ZigBee® PRO protocol stacks. The ZigBee® IP protocol stack is not supported at this time.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.2 ZigBee® SAS startup (class_id = 101, version = 0)

4.15.2.1 Overview

NOTE 1 In the specification of the ZigBee® COSEM ICs, the length of the octet-strings is indicated for information.

Instances of this IC are used to configure a ZigBee® PRO device with information necessary to create or join the network. The functionality that is driven by this object and the effect on the network depends on whether the object is located in a ZigBee® coordinator or in another ZigBee® device.

NOTE 2 The "ZigBee® SAS startup" object reflects the state of the ZigBee® HAN to the WAN (or a diagnostic tool connected to a ZigBee® connected DLMS®/COSEM Server). For example, if the object is in a Comms hub, then the attribute can be read out to show that the ZigBee® HAN has not been commissioned. The main function of this object is to provide information about a ZigBee® network to a client on the WAN (or via the optical port). The reason that there are multiple instances of the "ZigBee® SAS startup" object in Figure 29 is to express the idea that there may be multiple ZigBee® networks operated from a single Comms Hub.

ZigBee® SAS startup		**0...n**	**class_id = 101, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. short_address	(dyn.)	long-unsigned			0xFFFF	x + 0x08
3. extended_pan_id	(dyn.)	octet-string			0	x + 0x10
4. pan_id	(dyn.)	long-unsigned			0xFFFF	x + 0x18
5. channel_mask	(dyn.)	double-long-unsigned			0	x + 0x20
6. protocol_version	(static)	unsigned	0x02		0x02	x + 0x28
7. stack_profile	(static)	enum			0x02	x + 0x30
8. start_up_control	(dyn.)	unsigned			2	x + 0x38
9. trust_center_address	(dyn.)	octet-string			0	x + 0x40
10. link_key	(dyn.)	octet-string			0	x + 0x48
11. network_key	(dyn.)	octet-string			0	x + 0x50
12. use_insecure_join	(static)	boolean			FALSE	x + 0x58
Specific methods		***m/o***				

4.15.2.2 Attribute description

4.15.2.2.1 logical_name

Identifies the "ZigBee® SAS startup" object instance. See 6.2.29.

4.15.2.2.2 short_address

Defines the 16-bit address by which this device is known locally on the ZigBee® network. Value 0x0000 is given to a coordinator / Trust Center device, and only to these devices.

NOTE This short address will be issued to the device by the coordinator when the device joins the network.

4.15.2.2.3 extended_pan_ID

Defines the unique address by which the network is known externally.

The length of the octet-string is 8 octets.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.2.2.4 pan_ID

Defines the 16-bit address by which network is known locally.

4.15.2.2.5 channel_mask

Defines (as a bit mask) the set of radio channels which this PAN is permitted to use. Actual usage of channels within this set is determined by the coordinator on the basis of local conditions.

The mask is as defined in the ZigBee® specification.

4.15.2.2.6 protocol_version

Defines the version of the ZigBee® protocol to be used on the network.

4.15.2.2.7 stack_profile

Identifies the capabilities of the ZigBee ® stack.

enum:

(1) ZigBee®,
(2) ZigBee® PRO

4.15.2.2.8 start_up_control

Indicates the commissioning state of the ZigBee® device:

2: un-commissioned. Indicates that the device will seek to join the network if/when it is established;

0: commissioned. Indicates that the device should consider itself a part of the network indicated by the extended_pan_id attribute. In this case it will not perform any explicit join or rejoin operation.

4.15.2.2.9 trust_center_address

Defines the unique extended address of the Trust Center used for this network. This is often, but not necessarily, the address of the ZigBee® coordinator.

The length of the octet-string is 8 octets.

4.15.2.2.10 link_key

Defines the key value used to secure point-to-point communications between particular devices within the Smart Energy Profile. The way in which the raw key value is protected is not defined in this specification.

This is the APS link key or the PCLK. It's down to the implementation if this attribute is Hashed or in the clear or even if this is used.

If this is the Trust Center then this is not usually implemented.

This is usually read only, but may need to be writable for testing.

The length of the octet-string is 16 octets.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.2.2.11 network_key

Defines the key value used to secure general communications between devices. The way in which the raw key value is protected is not defined in this specification. The network key value may be set internally by the coordinator.

It is down to the implementation if this attribute is Hashed or in the clear or even if this is used.

If this is the TC then this is not usually implemented.

This is read only.

The length of the octet-string is 16 octets.

4.15.2.2.12 use_insecure_join

Indicates whether the coordinator is permitted to allow insecure joining as defined by ZigBee® specification.

4.15.3 ZigBee® SAS join (class_id = 102, version = 0)

4.15.3.1 Overview

Instances of this IC configure the behaviour of a ZigBee® PRO device on joining or loss of connection to the network. “ZigBee® SAS join” objects are present in all devices where a DLMS®/COSEM server controls the ZigBee® Radio behaviour, but it is not used when a device is acting as coordinator (as the coordinator device creates rather than joins a network). “ZigBee® SAS join” objects can be factory configured, or configured using another communications technique – e.g. an optical port.

ZigBee® SAS join	**0...n**	**class_id = 102, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. scan_attempts (static)	unsigned			3	x + 0x08
3. time_between_scans (static)	long-unsigned			1	x + 0x10
4. rejoin_interval	long-unsigned			60	x + 0x18
5. rejoin_retry_interval (static)	long-unsigned			900	x + 0x20
Specific methods	***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.3.2 Attribute description

4.15.3.2.1 logical_name

Identifies the “ZigBee® SAS join” object instance. See 6.2.29.

4.15.3.2.2 scan_attempts

Defines the number of consecutive scans that a ZigBee® device will perform on each attempt to rejoin a network, in the event of losing contact.

Remarks on usage

At boot time or when instructed to join a network, the device should complete up to three (3) (Note: as per the default) scan attempts to find a ZigBee® coordinator or router with which to associate.

If a device has not been commissioned, this means that when the user presses a button or uses another methodology to instruct the device to join a network, it will scan all of the channels up to three times (as per the default) to find a network that allows joining. If it has already been commissioned, it should scan up to three times (Note: as per the default) to find its original PAN to join. (ZigBee® Pro devices should try to find a network using their original extended PAN ID and ZigBee® devices can only try to find a network using their original PAN ID).

4.15.3.2.3 time_between_scans

Defines the period in seconds between consecutive scans in a single attempt to rejoin a network.

4.15.3.2.4 rejoin_interval

Defines the period in seconds that the device should wait after apparently becoming disconnected from a network, before attempting to rejoin.

4.15.3.2.5 rejoin_retry_interval

Defines the period in seconds for which the device should wait after a failed attempt to rejoin a network before trying again.

Remark on the *rejoin_retry_interval*

Imposes an upper bound on the *rejoin_interval* parameter – this is restarted if device is touched by human user, i.e. by a button press. This parameter is intended to restrict how often a device will scan to find its network in case the network is no longer present and therefore a scan attempt by the device would always fail (i.e., if a device finds it has lost network connectivity, it will try to re-join the network, scanning all channels if necessary). If the scan fails to successfully re-join, the device will wait for 15 min before attempting to re-join again. To be network friendly, it would be recommended to adaptively extend this time period if successive re-joins fail. It would also be recommended the device should try a re-join when triggered (via a control, button, etc.) and fall back to the *rejoin_retry_interval* if attempts to re-join fail again.

4.15.4 ZigBee® SAS APS fragmentation (class_id = 103, version = 0)

4.15.4.1 Overview

Instances of this IC configure the fragmentation feature of ZigBee® PRO transport layer. This fragmentation is not of concern to COSEM; the object merely allows configuration of the fragmentation function by an external manager (DLMS®/COSEM client).

Instances of this IC are present in all devices where a DLMS®/COSEM server controls the ZigBee® Radio behaviour.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

ZigBee® SAS APS fragmentation		0...n	class_id = 103, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. aps_interframe_delay	(static)	long-unsigned			50	x + 0x08
3. aps_max_window_size	(static)	long-unsigned			1	x + 0x10
Specific methods		***m/o***				

4.15.4.2 Attribute description

4.15.4.2.1 logical_name

Identifies the “ZigBee® SAS APS fragmentation” object instance. See 6.2.29.

4.15.4.2.2 aps_interframe_delay

Defines the delay in milliseconds between sending two blocks of a fragmented transmission.

4.15.4.2.3 aps_max_window_size

Defines the maximum number of unacknowledged frames that can be transmitted consecutively.

NOTE These initial values will allow for installation, and setting these values will depend on the specification of the ZigBee® Interface.

4.15.5 ZigBee® network control (class_id = 104, version = 0)

4.15.5.1 Overview

There will be a single instance of the “ZigBee® network control” IC in any device that can act as a ZigBee® coordinator controlled by the DLMS®/COSEM client. This class allows interaction between a DLMS®/COSEM client (head-end system) and a ZigBee® coordinator at times such as when the installation is commissioned.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

ZigBee® network control		0..1	class_id = 104, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. enable_disable_joining		boolean			FALSE	x + 0x08
3. join_timeout	(static)	long-unsigned			60	x + 0x10
4. active_devices	(dyn.)	array				x + 0x18
Specific methods		***m/o***				
1. register_device (data)		m				x + 0x20
2. unregister_device (data)		m				x + 0x28
3. unregister_all_devices (data)		o				x + 0x30
4. backup_PAN (data)		o				x + 0x38
5. restore_PAN (data)		o				x + 0x40
6. identify_device (data)		o				x + 0x48
7. remove_mirror (data)		o				x + 0x50
8. update_network_key (data)		o				x + 0x58
9. update_link_key (data)		o				x + 0x60
10. create_PAN (data)		m				x + 0x68
11. remove_PAN (data)		m				x + 0x70

4.15.5.2 Attribute description

4.15.5.2.1 logical_name

Identifies the “ZigBee® network control” object instance. See 6.2.29.

4.15.5.2.2 enable_disable_joining

A flag controlling whether devices are allowed to join the ZigBee® network. The flag is normally FALSE (joining disabled). At certain times – when a new device is expected to join – the flag is set externally to TRUE and then remains set for the duration of the *join_timeout*, or until externally reset if that happens sooner.

4.15.5.2.3 join_timeout

Defines the time period in seconds during which the coordinator device will permit joining of new devices, following setting of the *enable_disable_joining* flag.

4.15.5.2.4 active_devices

This attribute shows all of the currently authorised devices within the ZigBee® PAN.

```
array        active_device

active_device::= structure
{
        mac_address:                    octet-string,
        status:                         bit-string,
        maxRSSI:                        integer,
        averageRSSI:                    integer,
        minRSSI:                        integer,
        maxLQI:                         unsigned,
        averageLQI:                     unsigned,
        minLQI:                         unsigned,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
        last_communication_date-time:    octet-string,
        number_of_hops:                  unsigned,
        transmission_failures:           unsigned,
        transmission_successes:          unsigned,
        application_version:             unsigned,
        stack_version:                   unsigned
}
```

The length of the mac_address is 8 octets.

The length of the status is 8 bits.

The Max/Average/Min values are taken over the last 24 h period for each device. Period is from 00:00:01 to 00:00:00. They are always historical i.e. they refer to the last day.

```
status::= bit-string[8]

Bit 0 =    Authorised on PAN
Bit 1 =    Actively reporting on PAN
Bit 2 =    Unauthorised on PAN but has reported
Bit 3 =    Authorised after swap-out
Bit 4 =    SEP Transmitting
Bit 4 =    Reserved
Bit 6 =    Reserved
Bit 7 =    Reserved
```

Other elements are detailed in the ZigBee® specification.

4.15.5.3 Method description

4.15.5.3.1 register_device (data)

This method is called externally to instruct the coordinator that a device should be added to the list of authorised devices. This method does not actually join the devices to the network, they are just authorised to join at some time in the future.

```
data::= structure
{
        ieee_address:            octet-string,
        key_type:                enum,
        key:                     octet-string,
        device_type:             enum,
}
```

Where:

– ieee_address holds the IEEE address of the device. The length of the octet-string is 8 octets;

– key_type determines the type of content of key.

```
enum:

(0)     Pre-configured Link Key,
(1)     Install code,
(2)…(255)  Reserved,
```

NOTE 1 Install Codes will be subject to agreement across the members who are implementing a ZigBee® network. Therefore the length of install codes, the use of a CRC, and how they are padded when transported in this method are subject to agreement as part of a project specific companion specification.

– key is a pre-configured link key or install code. This will depend on the device being authorised to join the network. Its maximum length is 16 octets,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– device_type is linked to the enumeration shown below so that the coordinator has a method of understanding the possible services that the joining device may require.

NOTE 2 For example, it is likely that an implementation would require Mirror function for Gas, Water and Heat Meters connected by ZigBee®. However, determination of which ZigBee® support is needed for which device will be dependent on the organisation designing the smart metering solution, perhaps a government or a large utility.

device-type::= enum

(0) Electricity Meter
(1) Gas Meter
(2) Water Meter
(3) Thermal Meter
(4) Pressure Meter
(5) Heat Meter
(6) Cooling Meter
(7) Electric Vehicle charging Meter
(8) PV Generation Meter
(9) Wind Turbine Generation Meter
(10) Water Turbine Generation Meter
(11) Micro Generation Meter
(12) Solar Hot Water Generation Meter
(13) ZigBee® Controlled Load Switch
(14) ZigBee® Based Boost Button
(128) IHD
(129) Range extender
(130) CAD (Consumer Access Device)
(131) Thermostat
(132) Prepayment Terminal
(133) ZigBee® Controlled Load Switch
(134) ZigBee® Based Boost Button

4.15.5.3.2 unregister_device (data)

This method is called externally to instruct the coordinator that a device should leave the network.

data::= octet-string

It holds the ieee_address. The length of the octet-string is 8 octets.

4.15.5.3.3 unregister_all_devices (data)

This method is called externally to instruct the coordinator that all devices should leave the network.

data::= integer (0)

NOTE The likely use of this function is to ensure a network is empty of devices prior to destroying it.

4.15.5.3.4 backup_PAN (data)

This method instructs the coordinator to create a back-up of information that would be necessary to re-create the PAN.

NOTE 1 The storage location of the back-up is not currently defined and is an internal function of the DLMS®/COSEM server.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Method invocation return parameters are detailed below:

```
data::= structure

{
        date_time:                  octet-string,
        extended_PAN_ID:            octet-string,
        devices_to_backup:          array device_to_backup
}
```

Where:

– date-time refers to the time at which the backup process is due to begin. It is formatted as specified in 4.6.1;

– extended_PAN_ID identifies the PAN. The length of the octet-string is 8.

```
device_to_backup::= structure

{
        MAC_address:                octet-string,
        hashed_TC_link_key:   octet-string
}
```

Where:

– MAC_address hold the MAC address. The length of the octet-string is 8;

– the length of octet-string holding the hashed_TC_link_key is 16 octets.

NOTE 2 The method of hashing the link key is not part of this specification; it is defined in the ZigBee® Smart Energy Specification. MMO is currently used.

4.15.5.3.5 restore_PAN (data)

This method instructs the coordinator to restore a PAN using backup information. The storage location of the back-up is not currently defined and is an internal function of the DLMS®/COSEM Server.

```
data::= structure

{
        extended_PAN_ID:            octet-string,
        devices_to_restore:         array device_to_restore
}
```

Where:

– extended_PAN_ID identifies the PAN. The length of the octet-string is 8;

```
device_to_restore::= structure

{
        MAC_address:                octet-string,
        hashed_TC_link_key:         octet-string
}
```

Where:

– MAC_address holds the MAC address. The length of the octet-string is 8;

– the length of octet-string holding the hashed_TC_link_key is 16 octets.

NOTE The method of hashing the link key is not part of this specification; it is defined in the ZigBee® Smart Energy Specification. MMO is currently used.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.5.3.6 identify_device (data)

This method is called externally to instruct a device to identify itself to an engineer present on site, for example by sounding a buzzer.

data::= ieee_address

ieee_address: octet-string

ieee_address holds the IEEE address of the device. The length of the octet-string is 8 octets.

4.15.5.3.7 remove_mirror (data)

This method causes the removal of a ZigBee® mirror that reflects the real device identified by the mac_address parameter.

```
data::= structure
{
        mac_address:            octet-string,
        mirror_control:         bit-string
}
```

Where:

– MAC_address holds the MAC address. The length of the octet-string is 8;

– the mirror_control parameter is provided to support the execution of implementation-specific actions, which should be defined in a project specific companion specification.

EXAMPLE The following example is one of the possible options for the bit-string functionality.

0 = Force Gas Meter Removal.
NOTE Where a device removal is forced, the keys values for this device are removed; an APS ack from the device is not required.
1 = Clear all Mirror Data
2 = Clear Consumption registers / indexes
3 = Clear Demand & Max Demand registers
4 = Clear ZigBee® attributes
5 = Clear MPAN
6 = Clear Billing Information
7 = Clear Logs
8 = Clear OTA Firmware waiting
9..14 = Reserved
15 = Action all

Should the bit be set then the action is carried out.

The full meaning and actions that the DLMS®/COSEM server should undertake are implementation-specific actions, which should be defined in a project specific companion specification.

4.15.5.3.8 update_network_key (data)

This method requests that the ZigBee® coordinator updates the network key and propagate it to all devices on the PAN. For details of ZigBee® key management, see the ZigBee® specification.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.5.3.9 update_link_key (data)

This method requests that the ZigBee® coordinator updates the link key and propagate it to the identified device on the PAN. For details of ZigBee® key management, see the ZigBee® specification.

data::= ieee_address

ieee_address: octet-string

ieee_address holds the IEEE address of the device. The length of the octet-string is 8 octets.

4.15.5.3.10 create_PAN (data)

This method is called externally to instruct a coordinator to create a network using the configuration held in the ZigBee® SAS startup object.

data::= integer (0)

4.15.5.3.11 remove_PAN (data)

This method is called externally to instruct a coordinator to destroy a network by turning off the ZigBee® radio and removing all of the settings associated with the current PAN.

data::= integer (0)

4.15.6 ZigBee® tunnel setup (class_id = 105, version = 0)

4.15.6.1 Overview

A ZigBee® tunnel is established between two ZigBee® PRO devices to allow DLMS® APDUs to be transferred between them. The tunnel in effect extends WAN connectivity to ZigBee® devices not connected to the WAN through a ZigBee® device connected to the same ZigBee® network and connected to the WAN.

The ZigBee® tunnel setup objects would be present on the coordinator and on all other DLMS®/COSEM devices that are not connected to the WAN.

Creation of the tunnel is managed on demand and invisibly from the point of view of the DLMS®/COSEM client. The target device is implicitly identified by the COSEM addressing information.

NOTE See also the Gateway specification in IEC 62056-5-3:2023, Annex C.

ZigBee® tunnel setup		**0...n**	**class_id = 105, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. maximum_incoming_transfer_size	(static)	long-unsigned			0x05DC	x + 0x08
3. maximum_outgoing_transfer_size	(static)	long-unsigned			0x05DC	x + 0x10
4. protocol_address	(static)	octet-string length			0	x + 0x18
5. close_tunnel_timeout	(static)	long-unsigned			0xFFFF	x + 0x20
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.15.6.2 Attribute description

4.15.6.2.1 logical_name

Identifies the “ZigBee® tunnel setup” object instance. See 6.2.29.

4.15.6.2.2 maximum_incoming_transfer_size

Defines the maximum size, in octets, of the data packet that can be transferred to the tunnel client in the payload of a single **TransferData** (TransferData is a ZigBee® parameter) command. Behaviour on receipt of a larger packet is not defined in this specification.

4.15.6.2.3 maximum_outcoming_transfer_size

Defines the maximum size, in octets, of the data packet that can be sent from the tunnel client in the payload of a single **TransferData** command. Behaviour on transmission of a larger packet is not defined in this specification. See NOTEin 4.15.6.2.5.

NOTE The word *outcoming* is unusual but is adopted for commonality with ZigBee® specifications.

4.15.6.2.4 protocol_address

The *protocol_address* is implementation-specific, which should be defined in a project specific companion specification.

The length of the octet-string is 6 octets.

4.15.6.2.5 close_tunnel_timeout

Defines the time, in seconds that the ZigBee® server waits before closing an inactive tunnel on its own (without waiting for the *CloseTunnel* Command from the client within the ZigBee® specification) and freeing its resources

The timer is re-started with each reception of a command.

NOTE The word *outcoming* is unusual, but is adopted for commonality with ZigBee® specifications.

4.16 Interface classes for setting up and managing the DLMS®/COSEM profile for LPWAN networks

4.16.1 General

This subclause specifies ICs for setting up devices using the DLMS®/COSEM LPWAN communication profile and to diagnose the LPWAN network:

- the ICs specified in 4.16.2 are used to define setup and diagnostic objects for the SCHC-LPWAN part;
- the ICs specified in 4.16.3 are used to define setup and diagnostic objects for the lower layers.

For each specific LPWAN technology, a setup and a diagnostic IC should be specified. ICs for the LoRaWAN *1.0.3* technology are specified in 4.16.3.2.5.

4.16.2 Generic interface classes

4.16.2.1 SCHC-LPWAN setup (class_id = 126, version = 0)

4.16.2.1.1 Overview

Instances of this IC are available for configuring the parameters needed to set up a LPWAN device.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

SCHC-LPWAN setup		0..n	class_id = 126 , version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. lpwan_reference	(static)	octet-string				x + 0x08
3. schc_cd_rules	(static)	array				x + 0x10
4. schc_fr_param	(static)	structure				x + 0x18
Specific methods (if required)		***m/o***				

4.16.2.1.2 Attribute description

4.16.2.1.2.1 logical_name

Identifies the “LPWAN setup” object instance. See 6.2.23.

4.16.2.1.2.2 lpwan_reference

References an LPWAN technology specific setup object.

4.16.2.1.2.3 schc_cd_rules

Contains the necessary parameters to support LPWAN Static Context Header Compression (SCHC) compression and decompression rules

```
array schc_cd_rules_element

schc_cd_rules_element::= structure

{
    rule_id:                unsigned,
    field_descriptors:      array field_descriptor
}

field_descriptor::= structure

{
    field_id:               field_id_type,
    field_length:           unsigned,
    field_position:         unsigned,
    direction_indicator:    direction_indicator_type,

    target_value:           CHOICE

    {
        -- simple data types
        null-data                [0],
        boolean                  [3],
        bit-string               [4],
        double-long              [5],
        double-long-unsigned     [6],
        octet-string             [9],
        visible-string           [10],
        utf8-string              [12],
        bcd                      [13],
        integer                  [15],
        long                     [16],
        unsigned                 [17],
        long-unsigned            [18],
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
                long64                          [20],
                long64-unsigned                 [21],
                enum                            [22],
                float32                         [23],
                float64                         [24],
                date-time                       [25],
                date                            [26],
                time                            [27],

                -- complex data types
                array                           [1],
                structure                       [2],
                compact-array                   [19]
        }

        matching_operator:      matching_operator_type,
        compression_decompression_action:
        compression_decompression_action_type
}

field_id_type::= enum

{
        (0)             IPv6-version,
        (1)             IPv6-DiffServ,
        (2)             IPv6-FlowLabel,
        (3)             IPv6-Length,
        (4)             IPv6-NextHeader,
        (5)             IPv6-HopLimit,
        (6)             IPv6-Dev-Prefix,
        (7)             IPv6-DevIID,
        (8)             IPv6-AppPrefix,
        (9)             IPv6-AppIID,
        (10)            UDP-DevPort,
        (11)            UDP-AppPort,
        (12)            UDP-Length,
        (13)            UDP-Checksum,
        (14)            DLMS-Wrapper-version,
        (15)            DLMS-Wrapper-SSAP,
        (16)            DLMS-Wrapper-CSAP,
        (17)            DLMS-Wrapper-Length
}

direction_indicator_type::= enum
{
        (0)             Uplink,
        (1)             Downlink,
        (2)             bidirectional
}

matching_operator_type::= enum
{
        (0)             equal,
        (1)             ignore,
        (2)             MSBx,
        (3)             match-mapping
}

compression_decompression_action_type::= enum
{
        (0)             not-sent,
        (1)             value-sent,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
        (2)         mapping-sent,
        (3)         LSB,
        (4)         compute-length,
        (5)         compute-checksum,
        (6)         DevIID,
        (7)         AppIID,
}
```

4.16.2.1.2.4 schc_fr_param

Contains the necessary parameters to support LPWAN Static Context Header Compression (SCHC) fragmentation and reassembly parameters.

```
schc_fr_param::= structure

{
    rule_id_scheme:          enum
                              {
                                  (0)   fixed-size,
                                  (1)   variable-size
                              },
    max_packet_size:              long-unsigned,
    padding_l2_word_size:         unsigned,
    padding_bits_value:           unsigned,
    delay_after_transmission:     long-unsigned,
    interleaved_packet_tran:      boolean,
    window_size:                  unsigned,
    rule_params:                  array rule_param_element

}

rule_param_element::= structure
{
    rule_id:                 unsigned,
    rule_id_len:             unsigned,
    dtag_len:                unsigned,
    w_len:                   unsigned,
    fcn_len:                 unsigned,
    rcs_algorithm:           enum
                             {
                                  (0)   None,
                                  (1)   CRC32 using 0xEDB88320
                             },
    reliability_mode:        enum
                             {
                                  (0)   No-ACK,
                                  (1)   ACK-on-Error,
                                  (2)   ACK-Always
                             },
    retransmission_timer:    long-unsigned,
    inactivity_timer:        long-unsigned,
    max_ack_request:         unsigned
}
```

Where:

rule_id_scheme	The RuleID numbering scheme, fixed-size or variable-size Rule IDs, the way the Rule ID is transmitted;
max_packet_size	The maximum packet size that should ever be reconstructed by SCHC Decompression [octets];
padding_l2_word_size	The size of the L2 Word [bits];

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

padding_bits_value The value of the padding bits (0 or 1);

delay_after_transmission The delay to be added after transmission [milliseconds];

interleaved_packet_tran The support for interleaved packet transmission;

window_size The WINDOW_SIZE, for modes that use windows;

rule_id_len The length in bits for the RuleID header field of SCHC F/R message [bits];

dtag_len The length in bits for the Datagram Tag (DTag) header field of SCHC F/R message [bits];

w_len The length in bits for the W header field of SCHC F/R message [bits];

fcn_len The length in bits for the Fragment Compressed Number (FCN) header field of SCHC F/R message [bits];

rcs_algorithm The algorithm for the calculation of Reassembly Check Sequence (RCS) field of SCHC F/R message;

reliability_mode The reliability mode used;

retransmission_timer The Retransmission Timer duration [milliseconds];

inactivity_timer The Inactivity Timer duration [milliseconds];

max_ack_request The MAX_ACK_REQUEST value.

C/D rules defined for the DLMS® over SCHC profile are listed in Table 43.

Table 43 – C/D Rule 1

field_id	field_ length	field_ position	direction_ indicator	target_ value	matching_ operator	compression_ decompression_ action
IPv6-version	4	1	bidirectional	6	equal	not-sent
IPv6-DiffServ	8	1	bidirectional	0	equal	not-sent
IPv6-FlowLabel	20	1	bidirectional	0	equal	not-sent
IPv6-Length	16	1	bidirectional		ignore	comp-length
IPv6-NextHeader	8	1	bidirectional	17	equal	not-sent
IPv6-HopLimit	8	1	bidirectional	255	equal	not-sent
IPv6-Dev-Prefix	64	1	bidirectional	FE80::/64	equal	not-sent
IPv6-DevIID	64	1	bidirectional		ignore	DevIID
IPv6-AppPrefix	64	1	bidirectional	FE80::/64	equal	not-sent
IPv6-AppIID	64	1	bidirectional	1	ignore	not-sent
UDP-DevPort	16	1	bidirectional	4059	equal	not-sent
UDP-AppPort	16	1	bidirectional	4059	equal	not-sent
UDP-Length	16	1	bidirectional		ignore	comp-length
UDP-Checksum	16	1	bidirectional		ignore	comp-chk
DLMS-Wrapper-version	16	1	bidirectional	1	equal	not-sent
DLMS-Wrapper-SSAP	16	1	bidirectional	1	equal	not-sent
DLMS-Wrapper-CSAP	16	1	bidirectional	1	equal	not-sent
DLMS-Wrapper-Length	16	1	bidirectional		ignore	comp-length
NOTE If the device has to manage a second client, a second rule can be defined where the only field-id that is different will be DLMS-Wrapper-CSAP.						

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.16.2.2 SCHC-LPWAN diagnostic (class_id = 127, version = 0)

Instances of this IC are intended to provide diagnostic information regarding the transport of SCHC packets over the LPWAN network that can be useful at the application layer to investigate potential communications problems and service performance.

SCHC-LPWAN diagnostic		**0..n**	**class_id = 127 , version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name.	(static)	octet-string				x
2. schc_packets_tx_counter	(dynamic)	double-long-unsigned			0	x + 0x08
3. schc_packets_rx_counter	(dynamic)	double-long-unsigned			0	x + 0x10
4. schc_fragments_tx_counter	(dynamic)	double-long-unsigned			0	x + 0x18
5. schc_fragments_rx_counter	(dynamic)	double-long-unsigned			0	x + 0x20
6. schc_ack_tx_counter	(dynamic)	double-long-unsigned			0	x + 0x28
7. schc_ack_rx_counter	(dynamic)	double-long-unsigned			0	x + 0x30
Specific methods (if required)		***m/o***				
1. reset (data)		…				x + 0x40

4.16.2.3 Attribute description

4.16.2.3.1 logical_name

Identifies the “LPWAN Diagnostic” object instance. See 6.2.23.

4.16.2.3.2 schc_packets_tx_counter

It counts the SCHC non fragmented packet transmitted by the device.

4.16.2.3.3 schc_packets_rx_counter

It counts the SCHC non fragmented packet received by the device.

4.16.2.3.4 schc_fragments_tx_counter

It counts the SCHC fragments transmitted by the device.

4.16.2.3.5 schc_fragments_rx_counter

It counts the SCHC fragments received by the device.

4.16.2.3.6 schc_ack_tx_counter

It counts the ACK frame transmitted by the device.

4.16.2.3.7 schc_ack_rx_counter

It counts the ACK frame received by the device.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.16.2.4 Method description

4.16.2.4.1 reset (data)

This method resets all counters to 0.

data::= integer (0)

4.16.3 LPWAN specific interface classes

4.16.3.1 General

Interface classes specified in this clause are related to a given LPWAN technology. Currently, interface classes related to LoRaWAN *1.0.3* network are specified. ICs for other LPWAN technologies will be added later.

4.16.3.2 LoRaWAN® interface classes

4.16.3.2.1 General

These ICs define the LoRaWAN *1.0.3* parameters used during registration and deregistration process, needed at the device side.

NOTE 1 DEVEUI is defined using the MAC address setup interface class (class_id = 43).

NOTE 2 ABP (Activation by Personalization is not permitted).

4.16.3.2.2 LoRaWAN setup (class_id = 128, version = 0)

4.16.3.2.2.1 Overview

Instances of this interface class hold the parameters needed to set up a LoRaWAN *1.0.3* device.

This interface class mirrors the parametrization published in LoRa Alliance document: *LoRaWAN-TR006 DLMS® End Device Monitoring Guidelines.doc.*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

LoRaWAN® setup		0..n	class_id = 128 , version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. class	(dynamic)	enum				x + 0x08
3. state	(dynamic)	enum				x + 0x10
4. max_transmit_EIRP_setting	(static)	integer				x + 0x18
5. ADR_mode	(dynamic)	boolean				x + 0x20
6. regional_parameters	(dynamic)	enum				x + 0x28
7. device_operation	(dynamic)	structure				x + 0x38
8. modem_versions	(static)	structure				x + 0x40
9. devAddr	(dynamic)	double-long-unsigned				x + 0x48
10. join_strategy	(dynamic)	enum				x + 0x50
11. multicasts_parameters	(dynamic)	array				x + 0x58
Specific methods (if required)		***m/o***				
disconnect_from_network()		…				x + 060
change_class(data)		…				x + 068
change_region(data)		…				x + 0x70

4.16.3.2.3 Attribute description

4.16.3.2.3.1 logical_name

Identifies the “LoRaWAN setup” object instance. See 6.2.23.

4.16.3.2.3.2 class

Defines the class of the device.

Read only attribute, use change_class method to change the value.

enum:

(0) Class A,
(1) Class B,
(2) Class C

4.16.3.2.3.3 state

Shows the state of the network connection process of the device.

enum:

(0) not connected,
(1) connecting state (joining),
(2) not connected, connected at least one time,
(3) connected,
(4) connection error (join error)

4.16.3.2.3.4 max_transmit_EIRP_setting

Read only parameter. Maximum transmission capability of the device in dBm.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.16.3.2.3.5 ADR_mode

Adaptative **D**ata **R**ate is a LoRaWAN *1.0.3* parameter allowing the network server to control the device datarate and TX power. Recommended for static devices (devices expected to not roam between networks); it optimises energy consumption and network capacity.

boolean:

ADR_mode = TRUE: ADR is enabled

ADR_mode = FALSE: ADR is disabled

4.16.3.2.3.6 regional_parameters

Identifies the list of region names that a network and a device need to know to open a LoRaWAN service. Those regions are used to support local radio regulation specifics, like duty cycle (or LBT), max time over air per message, max power.

They also govern the selection of default radio channels, default radio settings, the channel mode (fixed or dynamic)

```
regional_parameters:= enum
{
        (0)   EU868,
        (1)   US915,
        (2)   CN779 (Deprecates 2021-01-01),
        (3)   EU433,
        (4)   AU915,
        (5)   CN470,
        (6)   AS923-1,
        (7)   AS923-2,
        (8)   AS923-3,
        (9)   KR920,
        (10)  IN865,
        (11)  RU864
}
```

4.16.3.2.3.7 device_operation

Counters that can help to identify issues.

```
device_operation::= structure
{
        TotalJoinRequestCounter:        long unsigned,
        TimeSinceLastJoinRequest:       long unsigned,
        TimeSinceLastJoinAccept:        long unsigned
}
```

Where:

– TotalJoinRequestCounter: Number of join requests since last reset.

– TimeSinceLastJoinRequest: Number of seconds elapsed since the device initiated the connection process.

– TimeSinceLastJoinAccept:Number of seconds elapsed since the device connected to the network.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.16.3.2.3.8 modem_versions

Identifies the modem hardware, software and protocol versions and the version of the regional parameters.

```
versions::= structure

{
        HardwareVersion: CHOICE
        {
                octet-string,
                visible-string
        },

        SoftwareVersion: CHOICE
        {
                octet-string,
                visible-string
        },

        RegionalParametersVersion: CHOICE
        {
                octet-string,
                visible-string
        },

        ProtocolVersion: CHOICE
        {
                octet-string,
                visible-string
        }
}
```

4.16.3.2.3.9 devAddr

32 bits device address identifying the device inside the current network. Assigned by the Network Server of the device. Should be 0 when not joined.

4.16.3.2.3.10 join_strategy

Defines the algorithm, timing/DR to apply during the joining.

To be documented by vendor. Vendor can assign a code to a Join strategy (where applying).

4.16.3.2.3.11 multicasts_parameters

Both DLMS®, IP and LoRaWAN are multicast technologies, LoRaWAN *1.0.3* supporting up to four multicast groups. One or more group(s) can be created for a set of DLMS® devices then SCHC gateway will forward all IP multicast data corresponding to that group to the end devices.

LoRa Alliance has published a recommended application layer messaging definition allowing to remotely manage devices groups, and program class B or C distribution window for a given group.

```
multicasts_parameters::= array multicast_parameters

multicast_parameters::= structure

{
        mc_addr:                    double-long-unsigned,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
                mc_key:                 octet-string,
                mc_min_fcount:          double-long-unsigned,
                mc_max_fcount:          double-long-unsigned,
                mc_start_time:          date-time,
                mc_duration:            long-unsigned,
                mc_class:               enum,
                mc_downlink_datarate:   unsigned-integer
                mc_downlink_frequency:  double-long-unsigned
            }
```

Where:

mc_addr: multicast device address;

NOTE 1 Should be 0 when multicast group is not used,32 bits.

mc_key: key used to encode the payload of the multicast group;

NOTE 2 128 bits.

mc_min_fcount: minimum multicast Fcounter, device should drop payload with lower FCount;

mc_max_fcount: maximum multicast Fcounter, device should drop payload with higher FCount

mc_start_time: Time to start the multicast session, if applicable

mc_duration: Maximum duration of the multicast session

mc_class: Device shall switch to the class during the length of the multicast window

enum:

Class B,
Class C
All others reserved

mc_downlink_datarate: Datarate of the multicast session

mc_downlink_frequency: Frequency of the multicast session, in Hz

An end-device supports a maximum of 4 simultaneous groups, and a minimum of 0. Up to 4 elements may be used in the array.

4.16.3.2.4 Method description

4.16.3.2.4.1 disconnect_from_network()

This method corresponds to a LoRaWAN *1.0.3* new join procedure.

Data::= integer (0)

4.16.3.2.4.2 change_class (data)

This method is used to change the class attribute. Data is an enum.

4.16.3.2.4.3 change_region (data)

This method is used to change locally the Regional Parameter. Can be used only in disconnected state.

4.16.3.2.5 LoRaWAN diagnostic (class_id = 129, version = 0)

4.16.3.2.5.1 Overview

Instances of this IC are intended to provide diagnostic information regarding the LoRaWAN *1.0.3* lower layers to investigate potential communications problems and service performance.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

LoRaWAN diagnostic			0..n	class_id = 129 , version = 0			
Attributes			***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1.	logical_name	(static)	octet-string				x
2.	internal_error_code	(dynamic)	enum				x + 0x08
3.	out_frames_u_counter	(dynamic)	double-long-unsigned			0	x + 0x10
4.	out_frames_c_counter	(dynamic)	double-long-unsigned			0	x + 0x18
5.	in_frames_u_counter	(dynamic)	double-long-unsigned			0	x + 0x20
6.	in_frames_c_counter	(dynamic)	double-long-unsigned			0	x + 0x28
7.	in_mac_command_ counter	(dynamic)	double-long-unsigned			0	x + 0x30
8.	in_mac_ans_error_c ounter	(dynamic)	double-long-unsigned			0	x + 0x38
9.	in_mac_ignored_ counter	(dynamic)	double-long-unsigned			0	x + 0x40
10.	in_per	(dynamic)	integer				x + 0x48
11.	in_mean_rssi_rx1	(dynamic)	integer				x + 0x50
12.	in_mean_snr_rx1	(dynamic)	integer				x + 0x58
13.	in_mean_rssi_rx2	(dynamic)	integer				x + 0x60
14.	in_mean_snr_rx2	(dynamic)	integer				x + 0x68
Specific methods (if required)			***m/o***				
1.	reset (data)		...				x + 0x70

4.16.3.2.5.2 Attribute description

4.16.3.2.5.2.1 logical_name

Identifies the “LPWAN Diagnostic” object instance. See 6.2.23.

4.16.3.2.5.2.2 internal_error_code

Fault condition identifier.

To be documented by vendor. When an end-device is faulty, vendor can assign a code to the fault condition.

4.16.3.2.5.2.3 out_frames_u_counter

Number of frames sent unconfirmed.

4.16.3.2.5.2.4 out_frames_c_counter

Number of frames sent confirmed.

4.16.3.2.5.2.5 in_frames_u_counter

Number of frames received unconfirmed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.16.3.2.5.2.6 in_frames_c_counter

Number of frames received confirmed.

4.16.3.2.5.2.7 In_mac_command_counter

Number of MAC command received from the network. Note: all commands are answered.

4.16.3.2.5.2.8 in_mac_ans_error_counter

Number of MAC answers replied with error.

4.16.3.2.5.2.9 in_mac_ignored_counter

Number of MAC commands ignored.

4.16.3.2.5.2.10 in_per

Per: packet error rate. Computed over the 5 received downlink.

Unit is percentage (56).

4.16.3.2.5.2.11 in_mean_rssi_rx1

Computed over the last 5 received downlink on rx1.

Unit is dBm.

4.16.3.2.5.2.12 in_mean_snr_rx1

Computed over the last 5 received downlink on rx1.

Unit is dB.

4.16.3.2.5.2.13 in_mean_rssi_rx2

Computed over the last 5 received downlink on rx2.

Unit is dBm.

4.16.3.2.5.2.14 in_mean_snr_rx2

Computed over the last 5 received downlink on rx2.

Unit is dB.

4.16.3.2.5.3 Method description

4.16.3.2.5.3.1 reset (data)

This method resets all counters to 0.

data:: = Integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17 Interface classes for setting up and managing the DLMS®/COSEM profile for Wi-SUN networks

4.17.1 Wi-SUN setup (class_id = 95, version 0)

4.17.1.1 Overview

This IC has the purpose of modelling the Wi-SUN network setup parameters.

All attributes in [FANSPEC] and [PHYSPEC] are written in camelCase and for the benefit of easy tracability these parameters are transposed into snake_case (underscore notation) within the DLMS®/COSEM specification.

Wi-SUN setup		**0..n**	**class_id = 95, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. network_name	(static)	octet-string				x + 0x08
3. routing_method	(dynamic)	enum	0	1	1	x + 0x10
4. pan_id	(dynamic)	long-unsigned	0	65534		x + 0x18
5. disc_imin	(static)	unsigned	1	255		x + 0x20
6. disc_imax	(static)	unsigned	1	8		x + 0x28
7. data_message_imin	(static)	unsigned	1	255	10	x + 0x30
8. data_message_imax	(static)	unsigned	1	8	3	x + 0x38
9. default_dio_interval_min	(static)	unsigned	1	255		x + 0x40
10. default_dio_interval_doublings	(static)	unsigned	1	8		x + 0x48
11. channel_plan	(static)	structure				x + 0x50
12. channel_function	(static)	enum	0	3		x + 0x58
13. excluded_channels	(static)	CHOICE				x + 0x60
14. join_state	(dynamic)	enum	1	5		x + 0x68
15. reg_channel_exclusions	(static)	structure				x + 0x70
Specific methods (if required)		***m/o***				
1. reset_join_state (data)		*o*				x + 0x78

4.17.1.2 Attribute description

4.17.1.2.1 logical_name

Identifies the “Wi-SUN setup” object instance. See 6.2.32.

4.17.1.2.2 network_name

Provides the name of the network that the device is allowed to join, or is a member of.

This is represented by ASCII characters.

NOTE The network name has a maximum of 32 octets.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17.1.2.3 routing_method

Provides the routing method employed by the device, two options are available:

enum:

(0) Layer 2 routing,
(1) Layer 3 routing

NOTE Default value according to [FANSPEC] is 1.

4.17.1.2.4 pan_id

Provides the PAN ID (configured at the Border Router and is known by every node on the network. This attribute is not controlled in the application layer.

NOTE 1 See [FANSPEC], 6.3.1.1.

NOTE 2 IEEE 802.15.4 reserves the value 65 535 for the broadcast PAN ID.

4.17.1.2.5 disc_imin

Provides the time in seconds of the time-out for network discovery.

NOTE Typical values of 60 s for large PANs (larger than 100 devices) or 15 s for small PANs (less than 100 devices) may be employed.

4.17.1.2.6 disc_imax

Provides the number of doublings of the value of "imin" (1-8 doublings of disc_imin).

NOTE Typical value of 4 for large PANs (approximately 16 min), value of 2 for small PANs (approximately 60 s).

4.17.1.2.7 data_message_imin

Provides the network broadcast timeout minimum value in seconds.

NOTE Typical default value of 10 s.

4.17.1.2.8 data_message_imax

Provides the maximum broadcast timeout as a function of data_message_imin (1-8 doublings of data_message_imin).

NOTE Typical default value of 3 (80 s).

4.17.1.2.9 default_dio_interval_min

Provides the ROLL RPL DIO minimum Interval (imin) The DIO trickle timer imin value is defined as 2 to the power of this value in ms (millseconds).

NOTE Typical default value for a large network: 19 (an imin of approximately 8 minutes) and a small network: 15 (an imin of approximately 32 s).

4.17.1.2.10 default_dio_interval_doublings

Provides the ROLL RPL DIO Interval IMAX and is a number of doublings of the value of default_dio_interval_min (1-8 doublings). A value of 1 for large PANs (approximately 16 minutes i.e. 8 x 2 min), value of 2 for small PANs (approximately 128 seconds i.e 32 x 2 x 2).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17.1.2.11 channel_plan

The channel plan sets the channel spacing, channel 0 frequency and number of channels plus the symbol rate.

Regional spectrum settings as specified in [PHYSPEC], Table 3. The values of regulatory_domain and operating_class uniquely identify the combination of operating parameters that the implementation is programmed to sustain. These parameters are subject to geographic regulation and may change from time to time depending on when regional spectrum authorities update their national frequency tables.

```
channel_plan::= structure
{
        regulatory_domain_identifier:    unsigned,
        operating_class_designator:      unsigned
}
```

Where:

– regulatory_domain_dentifier identifies the regulatory domain that the product is operating in, thereby identifying various PHY layer parameters including frequency bands, PHY modes, and channel spacing according to [PHYSPEC], Table 3.

– operating_class_designator identidfies the operating class of the device according to [PHYSPEC], Table 3. Operating class is unique only within the regulatory domain.

NOTE The above sets the channel spacing, channel 0 frequency and number of channels plus the symbol rate.

4.17.1.2.12 channel_function

Provides the channel function. There are currently 4 channel functions: fixed channel or a channel hopping mode where certain other parameters are specified as enumerated below.

enum:

Value	Description	Note
0	Fixed Channel	The transmitting node uses a single, fixed channel from the available channels.
1	TR51CF	The transmitting node uses the channel function described in ANSI/TIA-4957.200 (TR51CF), 7.1.
2	DH1CF	The transmitting node uses the Direct Hash channel function (DH1CF described in [FANSPEC], 6.3.4.5).
3	Vendor defined	The transmitting node uses a vendor defined channel function explicitly provided by the Hop Count and Hop List (defined in [FANSPEC] 6.3.2.3.2.1.4).
Other	reserved	

4.17.1.2.13 excluded_channels

Provides a means via which a node can have either a range (excluded_channel_ranges) or individual channels (excluded_channel_mask) notched out (not visited) of the node's channel hop sequence. Excluded_channel_ranges and excluded_channel_mask are configured for device specific purposes and shall be a subset of the channel_plan that fulfils regulatory channel restriction requirements. If no excluded channel ranges are set then the attribute shall be set to null-data.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Either excluded_channel_range OR excluded_channel_mask may be configured in a mutually exclusive way.

```
excluded_channels: CHOICE
{
        null-data [0],
        excluded_channel_ranges,
        excluded_channel_mask
}

null-data is selected when there are no excluded channels.

excluded_channel_ranges::= structure
{
        number_of_ranges:       integer,
        ranges_list:            array channel_range
}
        channel_range::= structure
        {
                start_channel_number:       long unsigned,
                end_channel_number:         long unsigned
        }
```

Where:

– number_of_ranges is the number of ranges in the ranges_list;

– ranges_list is an array of start and end channel numbers representing one or multiple excluded channel ranges;

– channel_range is a structure of start and end channel numbers that together represent an excluded channel range. The start and end channel shall also be considered to be included in the excluded range.

NOTE 1 Channels which are illegal or invalid may still not be used even if they are not included in a channel_range.

excluded_channel_mask::= bit-string

Excluded_channel_mask is a variable length bit_string (varying in multples of 8) representing the series of channels that are included or not included in the device's channel hop sequence. The Excluded Channel Mask field provides a means via which a node explicitly reports, for all channels within its channel hop sequence, which channels are and are not visited.

Bit	Channel
0 (first bit)	0
1	1
2	2
3	3
....	....
n	n

NOTE 2 [FANSPEC] states "If a bit in the mask is set to 1, the corresponding channel MUSTNOT be used. Bits that are not valid channels (unused following rounding to a multiple of 8) MUSTbe set to 1 on transmit and ignored on receipt."

This attribute is to be used for efficiently notching out individual channels rather than a range of channels within a band plan.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17.1.2.14 join_state

Provides the join state of the device.

enum:

(0) Initialisation state,
(1) Select PAN,
(2) Authenticate,
(3) Acquire PAN Config,
(4) Configure Routing,
(5) Operational,
(6)…(255) reserved

NOTE Applications on a device should only send or receive traffic when the device is in join state 5.

4.17.1.2.15 reg_channel_exclusions

Regional channel exclusions as designated in [FANSPEC]. The values of regulatory_domain and operating_class from channel_plan uniquely identify the combination of operating parameters that the implementation is programmed to sustain. This field identifies any specific channels that shall not be used when operating in a specific country.

```
reg_channel_exclusions::= structure
{
    num_channels:        long_unsigned,
    excluded_channels:   bit-string
}
```

Where:

num_channels — is defined as the total number of channels, according to [PHYSPEC], Table 3 when operating with the regulatory_domain and operating_class defined in channel_plan,

excluded_channels — is a bit-string, sized as num_channels, where the first bit, index 0 is Channel 0 and the last bit is “num_channels – 1”. For each index into excluded_channels, a value of “0” indicates the channel can be used and a value of “1” indicates the channel shall not be used.

NOTE Unlike the Channel Exclusion in the [FANSPEC], the regulatory channel exclusion is permanent and cannot be modified after configuration. These channels represent regulatory channel exclusions that apply to the particular country or region in that the configured device is deployed.

4.17.1.3 Method description

4.17.1.3.1 reset_join_state (data)

Resets join state attribute to 0.

4.17.2 Wi-SUN diagnostic (class_id = 96, version 0)

4.17.2.1 Overview

Instances of this IC are intended to provide diagnostic information regarding the Wi-SUN network that can be useful to investigate potential communication problems and service performance.

To facilitate management from the application, the following diagnostics are provided.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Wi-SUN diagnostics		0..n	class_id = 96, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. sysdescr	(dynamic)	octet-string				x + 0x08
3. ifnumber	(dynamic)	integer			1	x + 0x10
4. iftable	(dynamic)	array				x + 0x18
5. neighbour_table_information	(dynamic)	array				x + 0x20
6. transmission_information	(dynamic)	array				x + 0x28
Specific methods (if required)		***m/o***				
1. reset		*o*				x + 0x30

4.17.2.2 Attribute description

4.17.2.2.1 logical_name

Identifies the “Wi-SUN diagnostic” object instance. See clause 6.2.32.

4.17.2.2.2 sysdescr

A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contains printable ASCII characters.

4.17.2.2.3 ifnumber

The number of interfaces (ifentry) present within the physical device as per RFC 1213. Each interface shall have a value of ifentry defined in iftable.

4.17.2.2.4 iftable

A list of interface entries as per RFC 1213. The number of entries in the ifentry array is given by the value of attribute ifnumber; normally this shall be 1.

```
array ifentry

ifentry::=structure
{
        ifdescr                 octet-string,
        ifphysaddress           octet-string,
        ifinoctets              double-long-unsigned,
        ifinucastpackets        double-long-unsigned,
        ifinnucastpackets       double-long-unsigned,
        ifinerrors              double-long-unsigned,
        ifinunknownprotos       double-long-unsigned,
        ifoutoctets             double-long-unsigned,
        ifoutucastpackets       double-long-unsigned,
        ifoutnucastpackets      double-long-unsigned,
        ifouterrors             double-long-unsigned,
        ifoutQlen               double-long-unsigned,
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

Item	description
ifdescr	An ASCII description of the interface as per RFC 1213. A textual string containing information about the interface. This string should include the name of the manufacturer, the product name and the version of the hardware interface.
ifphysadress	The interface's address at the protocol layer immediately `below' the network layer in the protocol stack. For interfaces which do not have such an address (e.g., a serial line), this object should contain an empty octet string. In the case of Wi-SUN this entry shall be the EUI64.
ifinoctets	The total number of octets received on the interface, including framing characters.
ifinucastpackets	The number of subnetwork-unicast packets delivered to a higher-layer protocol.In the case of Wi-SUN diagnostic this shall be the number of packets passed to the DLMS®/COSEM application layer as per RFC 1213.
ifinnucastpackets	The number of non-unicast (i.e., subnetwork-broadcast or subnetwork-multicast) packets delivered to a higher-layer protocol as per RFC 1213.
ifinerrors	The number of inbound packets that contained errors preventing them from being deliverable to a higher layer protocol as per RFC 1213.
ifunknownprotos	The number of packets received via the interface at the network layer which were discarded because of an unknown or unsupported protocol as per RFC 1213.
ifoutoctets	The total number of octets transmitted out of the interface at the network layer, including framing characters as per RFC 1213.
ifoutucastpackets	The total number of packets that higher-level protocols requested be transmitted to a subnetwork-unicast address, including those that were discarded or not sent. In the case of Wi-SUN diagnostic higher layer protocols shall be defined as the DLMS®/COSEM application layer.
ifoutnucastpackets	The total number of packets that higher-level protocols requested be transmitted to a non-unicast (i.e., a subnetwork-broadcast or subnetwork-multicast) address, including those that were discarded or not sent. In the case of Wi-SUN diagnostic higher layer protocol shall be defined as the DLMS®/COSEM application layer as per RFC 1213.
ifouterrors	The number of outbound packets that could not be transmitted because of errors.
ifoutqlen	The length of the output packet queue (in packets) at the network layer,

4.17.2.2.5 neighbour_table_information

information The neighbour table keeps track of devices no more than 1 hop away.

```
array neighbour

neighbour::= structure
{
        neighbour_id:     octet-string,
        neighbour_type:   enum,
        device_rank:      unsigned,
        rssi:             long-unsigned,
        etx_to_parent:    double-long-unsigned
}
```

Where:

– neighbour_id is the EUI 64 of the neighbour device. The EUI-64 shall be of the modified EUI-64 format described in RFC 4291;

– neighbour_type is an enumeration defining whether the neighbour device is a parent, child or sibling.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

enum:

(0) Undetermined,
(1) Parent,
(2) Child,
(3) Sibling

– device_rank is a function of distance in hops of the device to the root device in the DODAG. The smaller the number the closer the device is to the DODAG root device;
– rssi is a measure of the signal strength of the neighbour;
– etx_to_parent is the Expected Number of transmissions required from the device to the destination via the parent in question required to transmit a packet. 1 is the perfect situation and FFFFFFFF is assumed to be equal to infinity and therefore a non-functioning link.

The first entry in the array of neighbours shall relate to the preferred parent and all subsequent entries shall be secondary parents.

NOTE Parent represents the node in the routing nearer to the Border Router (and may include the Border Router), child represents a node in the routing that has some dependency on this node, and sibling represents a node where communications is possible but is neither the parent or child of this node.

4.17.2.2.6 transmission_information

Provides transmission information and quality of service per interface.

```
array        transmission_element

transmission_element::= structure

{
        neighbour_id                octet-string,
        number_of_receptions        double-long-unsigned,
        number_of_sucessful_trans   double-long-unsigned,
        number_of_failed_trans              double-long-unsigned
}
```

Where:

– neighbour_id is the EUI-64 of the FAN interface. The EUI-64 shall be of the modified EUI-64 format described in RFC 4291;
– number_of_receptions is a simple counter of the total number of receptions from the device identified by interface_id;
– number_of_sucessful_trans is a simple counter of the total number of successful transmissions acknowledged by the device identified by interface_id;
– number_of_failed_trans is a simple counter of the total number of failed transmissions acknowledged by the device identified by interface_id.

4.17.2.3 Method description

4.17.2.3.1 reset (data)

Clears attributes 2…4 of the object instance, that is:

– sysdescr,
– ifnumber,
– iftable

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17.3 RPL diagnostic (class_id = 97, version 0)

4.17.3.1 Overview

As per [FANSPEC] 6.2.3.1.8 RPL is supported for the dissemination of IPv6 multicast packets. Instances of this IC are intended to provide diagnostic information regarding the Wi-SUN network from RPL diagnostic parameters as per RFC 6550 that can be useful to investigate potential communication problems and service performance issues.

RPL diagnostic		**0..n**	**class_id = 97, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. rpl_ instance_id	(dynamic)	unsigned	0		191	x + 0x08
3. dodag_version_number	(dynamic)	unsigned				x + 0x10
4. dodag_rank	(dynamic)	long				x + 0x18
5. grounded	(dynamic)	boolean				x + 0x20
6. mode_of_operation	(dynamic)	enum				x + 0x28
7. dodag_prf	(dynamic)	unsigned	0	7		x + 0x30
8. dodag_dtsn	(dynamic)	unsigned				x + 0x38
9. dodag_id	(dynamic)	octet-string				x + 0x40
Specific methods (if required)		***m/o***				
1. reset (data)		*o*				x + 0x48

4.17.3.2 Attribute description

4.17.3.2.1 logical_name

Identifies the “RPL diagnostic” object instance. See clause 6.2.32.

4.17.3.2.2 rpl_ instance_id

Holds the RPL instance ID as defined in RFC 6550. The values below are defined in RFC 6550 to identify whether or not the RPL Instance ID relates to a Global Instance ID, as in the case of coordinated networks or Local Instance ID as issued by a DODAG in the case of unilateral networks.

Bits 0 and 1 of the unsigned value act as flags where bit 0 when clear defines that the rpl_instance_id is global and has 127 posible values. Bit 0 set means that the rpl_instance_id is local, and in such case bit 1 shall always be clear assuming that the server is the recipient of messages from the rpl_instance_id allocating DODAG.

Instance ID Type	Start value	End value
Global Instance ID	0	127
Local Instance ID	128	191
Not Used	192	255

4.17.3.2.3 dodag_version_number

8-bit unsigned integer set by the DODAG root to the DODAGVersionNumber. RFC 6550, 8.2 describes the rules for DODAGVersionNumbers and how they affect DIO processing.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17.3.2.4 dodag_rank

Indicates the DODAG Rank of the node sending the DIO message. RFC 6550, 8.2 describes how Rank is set and how it affects DIO processing.

4.17.3.2.5 grounded

The Grounded 'G' flag indicates whether the DODAG advertised can satisfy the application-defined goal. If the flag is set, the DODAG is grounded. If the flag is cleared, the DODAG is floating.

4.17.3.2.6 mode_of_operation

The Mode of Operation (MOP) field identifies the mode of operation of the RPL instance as administratively provisioned at and distributed by the DODAG root. All nodes that join the DODAG shall honor the MOP in order to fully participate as a router, or else they shall only join as a leaf. MOP is encoded as detailed below:

enum:

(0) No Downward routes maintained by RPL,
(1) Non-Storing Mode of Operation,
(2) Storing Mode of Operation with no multicast support,
(3) Storing Mode of Operation with multicast support.

4.17.3.2.7 dodag_prf

Defines how preferable the root of this DODAG is compared to other DODAG roots within the instance. DAG Preference ranges from 0x00 (least preferred) to 0x07 (most preferred). The default is 0 (least preferred). RFC 6550, 8.2 describes how DAG Preference affects DIO processing.

4.17.3.2.8 dodag_dtsn

Destination Advertisement Trigger Sequence Number (DTSN) 8-bit unsigned integer set by the node issuing the DIO message. The Destination Advertisement Trigger Sequence Number (DTSN) flag is used as part of the procedure to maintain Downward routes. The details of this process are described in RFC 6550, 9.

4.17.3.2.9 dodag_id

128-bit IPv6 address represented as octet-string (length 16) set by a DODAG root that uniquely identifies a DODAG. The DODAGID shall be a routable IPv6 address belonging to the DODAG root.

4.17.3.3 Method description

4.17.3.3.1 reset (data)

Clears or (sets to default value) all dynamic fields of the object instance.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.17.4 MPL diagnostic (class_id = 98, version 0)

4.17.4.1 Overview

Instances of this IC are intended to provide diagnostic information regarding the Wi-SUN network from MPL diagnostic parameters as per RFC 7731 that can be useful at the application layer to investigate potential communication problems and service performance issues.

MPL diagnostic			**0..n**	**class_id =98 , version = 0**			
Attributes			***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1.	logical name	(static)	octet-string				x
2.	proactive_forwarding	(dynamic)	boolean				x + 0x08
3.	z	(dynamic)	unsigned	0	0	0	x + 0x18
4.	tunit	(dynamic)	unsigned	0x01	0xFE		x + 0x20
5.	se_lifetime	(dynamic)	long-unsigned	0x0001	0xFFFE		x + 0x28
6.	dm_k	(dynamic)	unsigned				x + 0x30
7.	dm_imin	(dynamic)	long-unsigned	0x0001	0xFFFE		x + 0x38
8.	dm_imax	(dynamic)	unsigned	0x01	0xFE		x + 0x40
9.	dm_t_exp	(dynamic)	long-unsigned	0x0001	0xFFFE		x + 0x48
10.	c_k	(dynamic)	unsigned				x + 0x50
11.	c_imin	(dynamic)	long-unsigned	0x0001	0xFFFE		x + 0x58
12.	c_imax	(dynamic)	unsigned	0x01	0xFE		x + 0x60
13.	c_t_exp	(dynamic)	long-unsigned	0x0001	0xFFFE		x + 0x68
Specific methods (if required)			***m/o***				
1.	reset (data)		o				x + 0x70

4.17.4.2 Attribute description

4.17.4.2.1 logical_name

Identifies the “MPL diagnostic” object instance. See 6.2.32.

4.17.4.2.2 proactive_forwarding

A flag to indicate PROACTIVE_FORWARDING.

NOTE This flag is set if PROACTIVE_FORWARDING = TRUE, as per RFC 7774, 2.1.

4.17.4.2.3 z

Reserved for future use. Defined as per RFC 7774, 2.1.

This is set to zero in all DHCP Servers, DHCP Clients shall ignore any none zero values.

4.17.4.2.4 tunit

Unit time of timer parameters (SE_LIFETIME and *_IMIN) in this option.

Values 0 and 0xFF are reserved and are not be used, as per RFC 7774, 2.1.

4.17.4.2.5 se_lifetime

SEED_SET_ENTRY_LIFETIME/TUNIT, in milliseconds (ms).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Values 0 and 0xFFFF are reserved and are not to be used, as per RFC 7774, 2.1.

4.17.4.2.6 dm_k

DATA_MESSAGE_K, as per RFC 7774, 2.1.

4.17.4.2.7 dm_imin

DATA_MESSAGE_IMIN/TUNIT, in milliseconds (ms).

Values 0 and 0xFFFF are reserved and are not to be used, as per RFC 7774, 2.1.

4.17.4.2.8 dm_imax

DATA_MESSAGE_IMAX. The actual maximum timeout is described as a number of doublings of DATA_MESSAGE_IMIN, as described in RFC 6206, 4.1.

Values 0 and 0xFF are reserved and shall not be used, as per RFC 7774, 2.1.

4.17.4.2.9 dm_t_exp

DATA_MESSAGE_TIMER_EXPIRATIONS, as per RFC 7774, 2.1.

Values 0 and 0xFFFF are reserved and are not to be used, as per RFC 7774, 2.1.

4.17.4.2.10 c_k

CONTROL_MESSAGE_K, as per RFC 7774, 2.1.

4.17.4.2.11 c_imin

CONTROL_MESSAGE_IMIN/TUNIT, in milliseconds (ms).

Values 0 and 0xFFFF are reserved and are not to be used, as per RFC 7774.

4.17.4.2.12 c_imax

CONTROL_MESSAGE_IMAX. The actual maximum timeout is described as a number of doublings of CONTROL_MESSAGE_IMIN.

Values 0 and 0xFF are reserved and are not to be used, as per RFC 7774.

4.17.4.2.13 c_t_exp

CONTROL_MESSAGE_TIMER_EXPIRATIONS.

Values 0 and 0xFFFF are reserved and are not to be used, as per RFC 7774.

4.17.4.3 Method description

4.17.4.3.1 reset (data)

Clears all dynamic fields of the object instance.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.18 Interface classes for setting up and managing the DLMS®/COSEM profile for ISO/IEC 14908 PLC networks

4.18.1 General

The interface classes specified here are used in devices implementing the IEC 62056-8-8:2020 DLMS®/COSEM communication profile for ISO/IEC 14908 series networks.

4.18.2 ISO/IEC 14908 identification (class_id = 130, version = 0)

4.18.2.1 Overview

Instances of the ISO/IEC 14908 identification IC allow the identification of the device and the network to which the device is connected.

ISO/IEC14908 identification		**0...n**	**class_id = 130, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. node_ID	(static.)	unsigned	1	127		x + 0x08
3. subnet_ID	(static)	unsigned	1	255		x + 0x10
4. domain_ID	(static)	octet-string				x + 0x18
5. program_ID	(static)	octet-string				x + 0x20
6. unique_node_ID	(static)	octet-string				x + 0x28
Specific methods		***m/o***				

4.18.2.2 Attribute description

4.18.2.2.1 logical_name

Identifies the “ISO/IEC 14908 identification” object instance. See 6.2.30.

4.18.2.2.2 node_ID

Identification of the node.

4.18.2.2.3 subnet_ID

Identification of the subnet to which the device is connected.

4.18.2.2.4 domain_ID

Identification of the network. It shall be 3 bytes.

4.18.2.2.5 program_ID

Uniquely identifies the functionalities and version of the firmware running on the device. It shall be 8 bytes.

4.18.2.2.6 unique_node_ID

Identification of the physical connection module connected to the ISO/IEC 14908-1:2012 network. Each compliant module is assigned a unique 48-bit identifier called Unique_Node_ID. This ID is unique worldwide and is set at the time of manufacture. The value of this identifier does not change from the time of manufacture.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.18.3 ISO/IEC 14908 protocol setup (class_id = 131, version = 0)

4.18.3.1 Overview

Instances of the ISO/IEC 14908 protocol setup IC allow the configuration of the ISO/IEC 14908 device.

ISO/IEC 14908 protocol setup		0...n	class_id = 131, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. inactivity_timeout	(static)	long-unsigned				x + 0x08
Specific methods		***m/o***				

4.18.3.2 Attribute description

4.18.3.2.1 logical_name

Identifies the “ISO/IEC 14908 protocol setup” object instance. See 6.2.30.

4.18.3.3 inactivity_timeout

Minutes of absence of adaptation layer messages from the NNAP before the LNAP is considered to be out of communication; see IEC 62056-8-8:2020, F.4.3.

4.18.4 ISO/IEC 14908 protocol status (class_id = 132, version = 0)

4.18.4.1 Overview

Instances of the ISO/IEC 14908 protocol status IC allow the status of the protocol in the ISO/IEC 14908 device to be determined.

ISO/IEC 14908 protocol status		0...n	class_id = 132, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. transmission_errors	(dyn)	long-unsigned				x + 0x08
3. transmit_tx_failure	(dyn)	long-unsigned				x + 0x10
4. transmit_tx_retries	(dyn)	long-unsigned				x + 0x18
5. receive_tx_full	(dyn)	long-unsigned				x + 0x20
6. lost_messages	(dyn)	long-unsigned				x + 0x28
7. missed_messages	(dyn)	long-unsigned				x + 0x30
8. layer2_received	(dyn)	long-unsigned				x + 0x38
9. layer3_received	(dyn)	long-unsigned				x + 0x40
10. messages_received	(dyn)	double-long-unsigned				x + 0x48
11. messages_validated	(dyn)	double-long-unsigned				x + 0x50
Specific methods		***m/o***				
1. reset (data)		o				X + 0x70

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.18.4.2 Attribute description

4.18.4.2.1 logical_name

Identifies the “ISO/IEC 14908 protocol status” object instance. See 6.2.30.

4.18.4.2.2 transmission_errors

This is a count of the number of transmission errors that have occurred on the network. A transmission error is detected via a CRC error during packet reception or as a packet that is less than 8 bytes long. This could result from a collision, a noisy medium, signal attenuation, etc.

See ISO/IEC 14908-1:2012, 13.8.2.

4.18.4.2.3 transmit_tx_failure

The number of times that the node failed to receive expected acknowledgments or responses after retrying the configured number of times.

See ISO/IEC 14908-1:2012, Clause B.8.

4.18.4.2.4 transmit_tx_retries

The number of retries sent by this node. This does not include retries used for messages sent with repeated service.

See ISO/IEC 14908-1:2012, Clause B.8.

4.18.4.2.5 receive_tx_full

The number of times that an incoming packet was discarded because there was no room in the transaction database.

See ISO/IEC 14908-1:2012, Clause B.8.

4.18.4.2.6 lost_messages

This is the number of messages that were addressed to the node that were thrown away because there was no application buffer available for the message.

See ISO/IEC 14908-1:2012, 13.8.2.

4.18.4.2.7 missed_messages

This is the number of messages that were on the network but could not be received because there was no network buffer (packet buffer) available for the message or the network buffer was too small to receive the message;

See ISO/IEC 14908-1:2012, 13.8.2.

4.18.4.2.8 layer2_received

The number of Layer-2 messages received by this node. Layer-2 messages are those that have correct CRC and can be addressed to any node.

See ISO/IEC 14908-1:2012, Clause B.8.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.18.4.2.9 layer3_received

The number of messages transmitted from layer 3 of the protocol processor. These can include network variable updates, explicit messages, acknowledgments, retries, reminders, service pin messages, and any other type of message.

See ISO/IEC 14908-1:2012, Clause B.8.

4.18.4.2.10 messages_received

Number of messages received, which are delivered to the Adaptation layer.

4.18.4.2.11 messages_validated

Number of correct messages received at Adaptation layer level.

4.18.4.3 Method description

4.18.4.3.1 reset (data)

Forces a reset of the object. By invoking this method, the values are set to the default value. The default value is an instance specific constant.

data::= integer (0)

NOTE IEC 62056-8-8:2020 specifies version = 0 which does not support the reset method.

4.18.5 ISO/IEC 14908 diagnostic (class_id = 133, version = 0)

4.18.5.1 Overview

Instances of the ISO/IEC 14908 diagnostic IC provides information about the device status inside the PLC network.

ISO/IEC 14908 diagnostic		**0…n**	**class_id = 133, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. plc_signal_quality_status	(dyn)	enum				x + 0x08
3. com_module_state	(dyn)	enum				x + 0x10
4. received_message_status	(dyn)	unsigned				x + 0x18
5. no_receive_buffer	(dyn)	long-unsigned				x + 0x20
6. transmit_no_data	(dyn)	long-unsigned				x + 0x28
7. backlog_overflows	(dyn)	long-unsigned				x + 0x30
8. late_ack	(dyn)	long-unsigned				x + 0x38
9. frequency_invalid	(dyn)	long-unsigned				x + 0x40
Specific methods		***m/o***				
1. reset (data)		o				X + 0x60

4.18.5.2 Attribute description

4.18.5.2.1 logical_name

Identifies the “ISO/IEC 14908 diagnostic” object instance. See 6.2.30.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.18.5.2.2 plc_signal_quality_status

Provides the PLC signal quality status:

enum:

(0) No PLC traffic detected,
(1) PLC traffic detected, but no traffic addressed to this device,
(2) A packet addressed to this device has been received. The signal margin is less than or equal to 9 dB,
(3) A packet addressed to this device has been received. The signal margin is between 12 dB or 15 dB,
(4) A packet addressed to this device has been received. The signal margin is greater than or equal to 18 dB,
(5) Meter is commissioned but its inactivity_timeout as defined in 4.18.1 has expired. The last packet addressed to this meter had a signal margin is less than or equal to 9 dB,
(6) Meter is commissioned but its inactivity_timeout as defined in 4.18.1 has expired. The last packet addressed to this meter had a signal margin at 12 dB or 15 dB,
(7) Meter is commissioned but its inactivity_timeout as defined in 4.18.1 has expired. The last packet addressed to this meter had a signal margin greater than or equal to 18 dB.

NOTE This parameter is updated every time a packet is received by the device.

4.18.5.2.3 com_module_state

Last received state of the device's power line IEC 62056-8-8:2020 communication module.

enum:

(0) Invalid state,
(1) Invalid state,
(2) Unconfigured,
(3) Applicationless,
(4) Configured,
(6) Hard-offline,
(12) Configured, soft offline,
(140) Configured, in bypass mode,
(255) No reply when status requested,
other Invalid states.

This field is updated on each device power-up, each re-synch request from the NNAP and each communication module reset. Any value which is not listed above is invalid state. See IEC 62056-8-8:2020,13.4.

NOTE Configured soft offline state (12) and Configured, in bypass mode state (140) are sub states of Configured state (4). See IEC 62056-8-8:2020, 13.4 and 13.8.

4.18.5.2.4 received_message_status

Unsigned integer, constructed as:

Bit 7 (MSB)	b6	b5	b4	b3	b2	b1	Bit 0 (LSB)
Transceiver phase			reserved				See below

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Transceiver phase bits are used as below:

	Bit 7	b6	Bit 5
No phase inversion present	Not set	Not set	Not set
In phase normal	Not set	Not set	Set
Plus 120 degrees inverted	Not set	Set	Not set
Minus 120 degrees normal	Not set	Set	Set
Phase 180 degrees inverted	Set	Not set	Not set
Plus 120 degrees normal	Set	Not set	Set
Minus 120 degrees inverted	Set	Set	Not set
Reserved	Set	Set	Set

4.18.5.2.5 no_receive_buffer

Number of times the IEC 62056-8-8:2020 lower layers detected a network inbound packet addressed to it, but has no buffer to store it. Hence the message could not be made available to the adaptation layer.

4.18.5.2.6 transmit_no_data

Number of times the adaptation layer attempted to transmit data to the IEC 62056-8-8:2020 stack but no buffer is available. Updated on occurrence. IEC 62056-8-8:2020,13.4.

4.18.5.2.7 backlog_overflows

Number of times the MAC layer predictive backlog counter overflowed. See IEC 62056-8-8:2020, Clause B.8.

4.18.5.2.8 late_ack

Number of times an ACK or response was received for a transaction that has been already completed. Typically, this occurs due to too short transaction timer values or network congestion.

4.18.5.2.9 frequency_invalid

Number of times the IEC 62056-8-8:2020 stack detects an invalid frequency. Updated on occurrence.

4.18.5.3 Method description

4.18.5.3.1 reset (data)

Forces a reset of the object. By invoking this method, the values of attributes 5..9 are set to the default value. The default value is an instance specific constant.

data::= integer (0)

NOTE IEC 62056-8-8:2020 specifies version = 0 which does not support the reset method.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5 Previous versions of interface classes

5.1 General

5.1.1 New versions of interface classes

Any modification of an existing IC affecting the transmission of service requests or responses results in a new version (version::= version+1) and shall be documented accordingly.

NOTE Previous versions of this document had additional text that applied a rule on the re-use and modification of attribute and method enumerators. This rule is no longer required as it placed an unnecessary constraint on interface classes with large numbers of attributes and/or methods.

Any modification of ICs will be recorded by moving the old version of an IC into the relevant clause below.

5.1.2 New interface classes

The DLMS UA reserves the right to be the exclusive administrator of interface classes.

5.1.3 Removal of interface classes

Besides one association object and the logical device name object no instantiation of an IC is mandatory within a meter. Therefore, even unused ICs will not be removed from the standard. They will be kept to ensure compatibility with possibly existing implementations.

5.2 Previous versions of interface classes – general

The subsequent subclauses list those IC specifications which were included in previous editions of this document. The previous IC versions differ from the current versions by at least one attribute and/or method and by the version number.

For new implementations in metering devices, only the current versions should be used.

Communication drivers at the client side should also support previous versions.

5.3 Previous versions of interface classes for parameters and measurement data

5.3.1 Profile generic (class_id = 7, version = 0)

5.3.1.1 Overview

The version listed here was valid in Edition 1 of IEC 62056-6-2 and replaced with effect from Edition 2.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Profile generic		0...n	class_id = 7, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. buffer	(dyn.)	array			x	x + 0x08
3. capture_objects	(static)	array				x + 0x10
4. capture_period	(static)	double-long-unsigned			x	x + 0x18
5. sort_method	(static)	enum			x	x + 0x20
6. sort_object	(static)	ObjectDefinition			x	x + 0x28
7. entries_in_use	(dyn.)	double-long-unsigned	0		0	x + 0x30
8. profile_entries	(static)	double-long-unsigned	1		1	x + 0x38
Specific methods		***m/o***				
1. reset data)						x + 0x58
2. capture (data)						x + 0x60
3. get_buffer_by_range (data)						x + 0x68
4. get_buffer_by_index (data)						x + 0x70

5.3.1.2 Attribute description

5.3.1.2.1 logical_name

Identifies the “Profile generic” object instance. For examples, see 6.2.19, 6.2.21, 6.2.46, etc.

5.3.1.2.2 buffer

The buffer attribute contains a sequence of entries. Each entry contains values of the captured objects (as returned by calling read (*current_value*)). The sequence is ordered according to the specified sort method. The buffer gets filled by subsequent calls to *capture ()*.

array entry

entry::= structure

Instance Specific

Default. The buffer is empty at installation.

NOTE Reading the entire buffer delivers only those entries which are “in use”

5.3.1.2.3 capture_objects

Specifies the list of capture objects (registers, clocks and profiles) that are assigned to this profile object. Upon a call of the *capture ()* service, the specified attributes of these objects are copied into the buffer of the profile.

```
array          ObjectDefinition
ObjectDefinition::= structure
{
  logical_name:     octet-string,
  class_id:         long-unsigned,
  attribute_index:  unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

where attribute_index is a pointer to the attribute within the object. attribute_index = 1 refers to the first attribute (i.e. *logical_name*), attribute_index = 2 to the 2nd, etc.)

5.3.1.2.4 capture_period

>= 1: Automatic capturing assumed. Specifies the capturing period in seconds.

0: no automatic capturing, capturing is trigger externally or capture events occur asynchronously.

5.3.1.2.5 sort_method

If the profile is unsorted, it works as a „first in first out“ buffer (it is hence sorted by capturing, and not necessarily by the time maintained in the clock object). If the buffer is full, the next call to *capture ()* will push out the first (oldest) entry of the buffer to make space for the new entry.

If the profile is sorted, a call to *capture ()* will store the new entry at the appropriate position in the buffer, moving all following entries and probably losing the least interesting entry. If the new entry would enter the buffer after the last entry and if the buffer is already full, the new entry will not be retained at all.

enum:

1: fifo,
2: lifo (last in first out),
3: largest,
4: smallest,
5: nearest_to_zero,
6: farest_from_zero
Default fifo

5.3.1.2.6 sort_object

If the profile is sorted, this attribute specifies the register or clock that the ordering is based upon.

ObjectDefinition see 5.3.1.2.3.

Default no object to sort by (only possible with *sort_method* = fifo or lifo).

5.3.1.2.7 entries_in_use

Counts the number of entries stored in the buffer. After a call of reset () the buffer does not contain any entries, and this value is zero. Upon each subsequent call of capture (), this value will be incremented up to the maximum number of entries that will get stored (see 5.3.1.2.8 *profile_entries*).

double-long-unsigned 0…profile_entries

Default 0

5.3.1.2.8 profile_entries

Specifies, how many entries should be retained in the buffer.

double-long-unsigned 1… (limited by physical size)

Default 1

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.3.1.3 Method description

5.3.1.3.1 reset (data)

Clears the buffer. The buffer has no valid entries afterwards; *entries_in_use* is zero after this call. This call does not trigger any additional operations of the capture objects, specifically, it does not reset any captured buffers or registers.

data = integer (0)

5.3.1.3.2 capture (data)

Copies the values of the objects to capture into the buffer by calling read (<*object_attribute*>) of each capture object. Depending on the *sort_method* and the actual state of the buffer this produces a new entry or a replacement for the less significant entry. As long as not all entries are already used, the *entries_in_use* attribute will be incremented.

This call does not trigger any additional operations within the capture objects such as *capture ()* or *reset ()*.

Note that only some attributes of the captured objects might be stored, not the complete object.

data = integer (0)

5.3.1.3.3 write (...)

Any write access to one of the attributes describing the static structure of the buffer will automatically call a *reset ()* and this call will propagate to all other profiles capturing this profile.

If writing to *profile_entries* is attempted with a value too large for the buffer, it will be set to the maximum possible value (restricted by physical size, typically laid down in the firmware).

5.3.1.3.4 get_buffer_by_range (data)

Reads all entries between the given limits.

restricting_ object	in	ObjectDefinition Defines the register or clock restricting the range of entries to be retrieved.
from_value	in	instance_specific Oldest or smallest entry to retrieve.
to_value	in	instance_specific Newest or largest entry to retrieve.
selected_values	in	List of columns to retrieve: array ObjectDefinition If the array is empty (has no entries), all captured data is returned. Otherwise, only the columns specified in the array are returned. The type *ObjectDefinition* is specified above (*Capture_Objects*)
entries	out	See 5.3.1.2.7 *entries_in_use* above.
array (of *instance_ specific_value)*	out	Sorted sequence of entries containing the requested values of the capture objects.
data::= structure {restricting_object; from_value; to_value; selected_values} In the response “data” is an array of instance_specific_value.		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.3.1.3.5 get_buffer_by_index (data)

Read all entries between the given limits.

from_index	in	double-long-unsigned First entry to retrieve.
to_index	in	double-long-unsigned Last entry to retrieve.
from_selected_ value	in	long-unsigned index of first value to retrieve
to_selected_value	in	long-unsigned index of last value to retrieve.
entries	out	See 5.3.1.2.7.
array of *instance_ specific_value*	out	Sorted sequence of entries containing the requested values of the capture objects.
data::= structure {from_index; to_index; selected_values}. In the response “data” is an array of instance_specific_value.		

5.3.2 Compact data (class_id = 62, version = 0)

5.3.2.1 Overview

Instances of the “Compact data” IC allow capturing the values of COSEM object attributes as determined by the *capture_objects* attribute. Capturing can take place:

- on an external trigger (explicit capturing); or
- upon reading the *compact_buffer* attribute (implicit capturing);

as determined by the *capture_method* attribute.

The values are stored in the *compact_buffer* attribute as an octet-string.

The set of data types is identified by the *template_id* attribute. The data type of each attribute captured is held by the *template_description* attribute.

The client can reconstruct the data in the uncompacted form – i.e. including the COSEM attribute descriptor, the data type and the data values – using the *capture_objects*, *template_id* and *template_description* attributes.

Compact data		**0...n**	**class_id = 62, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. compact_buffer	(dyn.)	octet-string				x + 0x08
3. capture_objects	(static)	array				x + 0x10
4. template_id	(static)	unsigned				x + 0x18
5. template_description	(dyn.)	octet-string				x + 0x20
6. capture_method	(static)	enum				x + 0x28
Specific methods		***m/o***				
1. reset (data)		o				x + 0x38
2. capture (data)		o				x + 0x40

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.3.2.2 Attribute description

5.3.2.2.1 logical_name

Identifies the “Compact data” object instance. See 6.2.41.

5.3.2.2.2 compact_buffer

Contains the values of the attributes captured as an *octet-string*.

When the data captured is of type *octet-string, bit-string, visible-string, utf8-string* or *array* the length is also included here.

5.3.2.2.3 capture_objects

Specifies the list of COSEM object attributes that are assigned to the “Compact data” object instance.

The *template_id* attribute shall be the first element in the *capture_objects* array.

Upon an explicit or implicit invocation of the *capture (data)* method the values of the selected attributes are captured into the *compact_buffer*.

```
array        capture_object_definition
capture_object_definition::= structure
{
        class_id:          long-unsigned,
        logical_name:      octet-string,
        attribute_index:   integer,
        data_index:        long-unsigned
}
```

Where:

– attribute_index is a pointer to the attribute within the object, identified by class_id and logical_name: attribute_index 1 refers to the 1st attribute (i.e. the *logical_name*), attribute_index 2 to the 2nd attribute, etc.; attribute_index 0 refers to all public attributes;

– data_index is a pointer selecting one or several specific elements of an attribute with a complex data type (structure or array).

- if the data type of the attribute is simple, then data_index has no meaning;
- if the data type of the attribute is a structure or an array, then data_index points to one or several specific elements in the structure or array;
- when the attribute is the *buffer* of a “Profile generic” object, the data_index carries selective access parameters.

data_index:	MS-Byte		LS-Byte
	Upper nibble	Lower nibble	

- 0x0000 = identifies the whole attribute;
- 0x0001 to 0x0FFF = identifies one element in the complex attribute. The first element in the complex attribute is identified by data_index 1;
- 0x1000 to 0xFFFF = selective access to the array holding the buffer of a “Profile generic” object. The data-index selects entries within a number of last (recent) time periods, or a number of last (recent) entries, as well as the columns in the array.

The encoding is specified in Table 16.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.3.2.2.4 template_id

Contains the identifier of the template.

It shall uniquely identify the instance of the “Compact data” IC and the *template_description*.

5.3.2.2.5 template_description

Provides the data type of each attribute captured. It is an *octet-string* generated automatically by the server upon the programming of the *capture_objects* and it has the following structure:

– the first octet is 0x02 (the tag of a structure);
– this is followed by the number of elements in the structure – the same as the number of elements in the *capture_objects* array – encoded as a variable length integer;
– this is followed by the data type of each attribute, in the same order as in the *capture_object* array:
  - in the case of attributes with simple data type, the data type is represented by a single octet, carrying the tag of the data type. In the case of bit-string [4], octet-string [9], visible-string [10], utf8-string [12] the length of the string is part of the data held in the *compact_buffer*;
  - in the case of an array [1], the data type is represented by a single octet 0x01. This is followed by the type description of the elements in the array. The number of elements in the array is part of the data held in the *compact_buffer*;
  - in the case of a structure [2], the data type is represented by a single octet 0x02, followed by the number of elements inside the structure followed by the tag of each element of the structure.

5.3.2.2.6 capture_method

Defines the way the *compact_buffer* is updated.

enum:

(0) Capture upon invoking the *capture (data)* method. This may occur remotely or locally (explicit capturing),
(1) Capture upon reading the *compact_buffer* attribute (implicit capturing).

5.3.2.3 Method description

5.3.2.3.1 reset (data)

Clears the *compact_buffer*. After invoking this method the *compact_buffer* holds an octet-string of 0 length, until a new capture takes place.

This call does not trigger any additional operations of the capture objects. Specifically, it does not reset any captured attributes.

data::= integer(0)

5.3.2.3.2 capture (data)

Copies the values of the attributes into the *compact_buffer* by reading each capture object.

This call does not trigger any additional operations within the capture objects such as capture () or reset ().

data::= integer(0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.3.2.4 Behaviour of the object after modification of certain attributes:

Any modification of the *capture_objects* resets the *compact_buffer* and automatically updates the *template_description*.

5.3.2.5 Restrictions

When defining the *capture_object* attribute, circular references shall be avoided.

5.4 Previous versions of interface classes for access control and management

5.4.1 Association SN (class_id = 12, version = 0)

5.4.1.1 Overview

The version listed here was valid in Edition 1 of IEC 62056-6-2 and it is replaced with effect from Edition 2.

Device Association View	**0..1** [a]	**class_id = 12, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. object_list (static)	objlist_type				x + 0x08
Specific methods	***m/o***				
1. getlist_by_classid (data)	o				x + 0x20
2. getobj_by_logicalname (data)	o				x + 0x28
3. read_by_logicalname (data)	o				x + 0x30
4. get_attributes&services (data)	o				x + 0x38
5. change_LLS_secret (data)	o				x + 0x40
6. change_HLS_secret (data)	o				x + 0x48
7. get_HLS_challenge (data)	o				x + 0x50
8. reply_to_HLS_challenge (data)	o				x + 0x58

5.4.1.2 Attribute description

5.4.1.2.1 logical_name

Identifies the type of the client-server association. See 6.2.33.

5.4.1.2.2 object_list

Contains the list of all objects with their short_name (DLMS ObjectName of the first attribute), class_id, version[b] and *logical_name*.

```
Objlist_type::=     array       objlist_element
objlist_element::= structure
{
        short_name:     long,
        class_id:       long-unsigned,
        version =       unsigned,
        logical_name:   octet-string
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.1.3 Method description

5.4.1.3.1 getlist_by_classid (data)

Delivers the subset of the object_list for a specific class_id.

data::= class_id: long-unsigned

For the response: data::= objlist_type

5.4.1.3.2 getobj_by_logicalname (data)

Delivers the entry of the object_list for a specific logical_name and class_id.

data::= structure
{
    class_id: long-unsigned,
    logical_name: octet-string
}
For the response: data::= objlist_element

5.4.1.3.3 read_by_logicalname (data)

Reads attributes for specific objects. The objects are specified by their class_id and their logical_name.

data::= arrayAttributeIdentification
AttributeIdentification::= structure
{
    class_id: long-unsigned,
    logical_name: octet-string,
    attribute_index: unsigned
}

where:

– attribute_index is a pointer (i.e. offset) to the attribute within the object.

– attribute_index = 0 delivers all attributes[c], attribute_index = 1 delivers the first attribute (i.e. logical_name), etc.

For the response: data is according to the type of the attribute.

5.4.1.3.4 get_attributes&services (data)

Delivers information about the access rights to the attributes and services within the actual association. The objects are specified by their class_id and their logical_name.

array ObjectIdentification

ObjectIdentification::= structure
{
    class_id: long-unsigned,
    logical_name: octet-string
}
For the response
data::= array AccessDescription

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
AccessDescription::= structure
{
        read_attributes:   bit-string,
        write_attributes:  bit-string,
        services:          bit-string
}
```

The position in the bit-string identifies the attribute/service (first position ↔ first attribute, first position ↔ first service) and the value of the bit specifies whether the attribute/service is available (bit set) or not available (bit clear).

5.4.1.3.5 change_LLS_secret (data)

Changes the LLS secret (for example password).

data::= octet-string new LLS secret

5.4.1.3.6 change_HLS_secret (data)

Changes the HLS secret (for example encryption key).

data::= octet-string[d] new HLS secret

5.4.1.3.7 get_HLS_challenge (data)

Asks the server for the client “challenge” (for example random number).

data::= octet-string client challenge

5.4.1.3.8 reply_to_HLS_challenge (data)

Delivers the “secretly” processed “challenge” back to the server.

data::= octet-string client’s response to the challenge

If the authentication is accepted, then the response is successful [0]. If the authentication is not accepted, then the response is set to data-access-error [1].

NOTE 1 Per client server association.

NOTE 2 Within an client-server association, there are never two objects with the same class_id and logical_name differing only in the versions.

NOTE 3 If at least one attribute has no read access right under the current association, then a read_by_logicalname () to attribute index = 0 reveals the error message “scope-of-access-violated” (comp. IEC 61334-4-41:1996, Clause A.12 .

NOTE 4 The structure of the “new secret” depends on the security mechanism implemented. The “new secret” may contain additional checkbits and it may be encrypted.

5.4.2 Association SN (class_id = 12, version = 1)

5.4.2.1 Overview

This IC allows modelling of AAs between a server and a client, using the application context *short name referencing.* A COSEM logical device may have one instance of this IC for each association the device is able to support.

The **short_name** of the current “Association SN” object itself is fixed within the COSEM context. It is given in 4.1.3 as 0xFA00.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Association SN		0...n	class_id = 12, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. object_list	(static)	objlist_type				x + 0x08
Specific methods		***m/o***				
1. reserved from previous versions		o				
2. reserved from previous versions		o				
3. read_by_logicalname (data)		o				x + 0x30
4. get_attributes&methods (data)		o				x + 0x38
5. change_LLS_secret (data)		o				x + 0x40
6. change_HLS_secret (data)		o				x + 0x48
7. reserved from previous versions						
8. reply_to_HLS_authentication (data)		o				x + 0x58

5.4.2.2 Attribute description

5.4.2.2.1 logical_name

Identifies the “Association SN” object instance. See 6.2.33.

5.4.2.2.2 object_list

Contains the list of all objects with their base_names (short_name), class_id, version and *logical_name*. The base_name is the DLMS objectName of the first attribute (*logical_name*).

```
objlist_type::=      array objlist_element

objlist_element::=   structure
{
        base_name:       long,
        class_id:        long-unsigned,
        version =        unsigned,
        logical_name:    octet-string
}
```

selective access (see 4.1.4) to the attribute object_list may be available (optional). The selective access parameters are as defined in Table 44.

Table 44 – Parameters for selective access to the object_list attribute

Access selector value	Parameter	Comment
1	class_id: long-unsigned	Delivers the subset of the object_list for a specific class_id. For the response: data::= objlist_type
2	structure { class_id: long-unsigned, logical_name: octet-string }	Delivers the entry of the object_list for a specific class_id and logical_name. For the response: data::= objlist_element

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.2.3 Method description

5.4.2.3.1 Reserved from previous version

This method is reserved.

5.4.2.3.2 Reserved from previous version

This method is reserved.

5.4.2.3.3 read_by_logicalname (data)

Reads attributes for specific objects. The objects are specified by their class_id and their logical_name. With this method, the parameterized access feature can also be used.

```
data::= array      attribute_Identification

attribute_Identification::= structure
{
      class_id:          long-unsigned,
      logical_name:      octet-string,
      attribute_index:   integer
}
```

Where:

– attribute_index is a pointer (i.e. offset) to the attribute within the object;

– attribute_index = 0 delivers all attributes, attribute_index = 1 delivers the first attribute (i.e. *logical_name*), etc.).

For the response: data is according to the type of the attribute.

NOTE If at least one attribute has no read access right under the current association, then a read_by_logicalname () to attribute index 0 reveals the error message “scope of access violation” (see IEC 61334-4-41:1996, Clause A.12).

5.4.2.3.4 get_attributes& methods (data)

Delivers information about the access rights to the attributes and methods within the actual association. The objects are specified by their class_id and their *logical_name*. With this method, the parameterized access feature can also be used.

```
data::= array      object_identification

object_identification::= structure
{
      class_id:          long-unsigned,
      logical_name:      octet-string
}

For the response
data::= array      access_description

access_description::= structure
{
      read_attributes:          bit-string,
      write_attributes:         bit-string,
      methods:                  bit-string
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The position in the bit-string identifies the attribute/method (first position ↔ first attribute, first position ↔ first method) and the value of the bit specifies whether the attribute/method is available (bit set) or not available (bit clear).

Depending on the implementation, some attributes or methods of some objects may not be needed. In this case, such attributes or methods may not be accessible (neither read access, nor write access to attributes, no access to methods).

5.4.2.3.5 change_LLS_secret (data)

Changes the LLS secret (for example password).

data::= octet-string new LLS secret .

5.4.2.3.6 change_HLS_secret (data)

Changes the HLS secret (for example encryption key).

data::= octet-string new HLS secret

NOTE The structure of the “new secret” depends on the security mechanism implemented. The “new secret” may contain additional check bits and it may be encrypted.

5.4.2.3.7 Reserved from previous version

This method is reserved.

5.4.2.3.8 reply_to_HLS_ authentication (data)

The remote invocation of this method delivers the client's “secretly” processed “challenge StoC” (f(StoC)) back to the server as the data service parameter of the invoked Write.request service.

data::= octet-string client’s response to the challenge

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

5.4.3 Association SN (class_id = 12, version = 2)

5.4.3.1 Overview

COSEM logical devices able to establish AAs within a COSEM context using SN referencing, model the AAs using instances of the “Association SN” IC. A COSEM logical device may have one instance of this IC for each AA the device is able to support.

The **short_name** of the “Association SN” object itself is fixed within the COSEM context. See 4.1.3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Association SN		0...n	class_id = 12, version = 2			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. object_list	(static)	objlist_type				x + 0x08
3. access_rights_list	(static)	access_rights_type				x + 0x10
4. security_setup_reference	(static)	octet-string				x + 0x18
Specific methods		***m/o***				
1. *reserved from previous versions*		o				
2. *reserved from previous versions*		o				
3. read_by_logicalname (data)		o				x + 0x30
4. *reserved from previous versions*		o				
5. change_secret (data)		o				x + 0x40
6. *reserved from previous versions*		o				
7. *reserved from previous versions*						
8. reply_to_HLS_authentication (data)		o				x + 0x58

5.4.3.2 Attribute description

5.4.3.2.1 logical_name

Identifies the “Association SN” object instance. See 6.2.33.

5.4.3.2.2 object_list

Contains the list of all objects with their base_name (short_name), class_id, version and *logical_name*. The base_name is the DLMS objectName of the first attribute (*logical_name).*

```
objlist_type::= array            objlist_element
objlist_element::= structure
{
    base_name:      long,
    class_id:       long-unsigned,
    version:        unsigned,
    logical_name:   octet-string
}
```

selective access (see 4.1.4) to the attribute object_list may be available. The access selector values and their parameters are as defined in Table 45.

5.4.3.2.3 access_rights_list

Contains the access rights to attributes and methods.

The link between the *object_list* and the *access_rights_list* is the base_name, present in both the objlist_element structure and the access_right_element structure. Therefore, the base_names on the two lists shall be the same. The number – and preferably, the order – of the elements in the array of objlist_element and the array of access_right_element shall also be the same.

```
access_rights_type::= array          access_rights_element
access_rights_element::= structure
{
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
        base_name:              long,
        attribute_access:       attribute_access_descriptor,
        method_access:          method_access_descriptor
}
attribute_access_descriptor::= array attribute_access_item
 attribute_access_item::= structure
{
 attribute_id:              integer,
 access_mode:     enum:
                            (0)    no_access,
                            (1)    read_only,
                            (2)    write_only,
                            (3)    read_and_write,
                            (4)    authenticated_read_only,
                            (5)    authenticated_write_only,
                            (6)    authenticated_read_and_write

 access_selectors:          CHOICE
 {
        null-data           [0],
        array integer       [1]
 }
}
method_access_descriptor::=array        method_access_item


method_access_item::= structure
{
        method_id:              integer,
        access_mode:            enum:
                    (0)    no_access,
                    (1)    access,
                    (2)    authenticated_access
}
```

selective access (see 4.1.4) to the attribute *access_rights_list* may be available (optional). The access selector values and their parameters are as defined in Table 45.

5.4.3.2.4 security_setup_ reference

References the “Security setup” object by its *logical_name*. The referenced object manages security for a given “Association SN” object instance.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 45 – Parameters for selective access to the *object_list* and *access_rights_list* attribute

Access selector value	Parameter	Available with attribute	Comment
1	class_id: long-unsigned	2	Delivers the subset of the object_list for a specific class_id. For the response: data::= objlist_type

2	structure
	{
	class_id: long-unsigned,
	logical_name: octet-string
	}
		2	Delivers the entry of the object_list for a specific class_id and *logical_name*.
			For the response: data::= objlist_element

3	base_name: long	2, 3	In the case of attribute 2, delivers the entry of the object_list for a specific base_name. For the response: data::= objlist_element In the case of attribute 3, delivers the entry of the access_rights_list for a specific base_name. For the response: data::=access_rights_element

5.4.3.3 Method description

5.4.3.3.1 Reserved from previous version

This method is reserved.

5.4.3.3.2 Reserved from previous version

This method is reserved.

5.4.3.3.3 read_by_logicalname (data)

Reads attributes for selected objects. The objects are specified by their class_id and their logical_name. With this method, the parameterized access feature can also be used.

```
data::= array      attribute_identification

attribute_identification::= structure
{
      class_id:          long-unsigned,
      logical_name:      octet-string,
      attribute_index:   integer
}
```

Where:

– *attribute_index* is a pointer (i.e. offset) to the attribute within the object;

– attribute_index 0 delivers all attributes; attribute_index 1 delivers the first attribute (i.e. *logical_name*), etc.).

For the response: data is according to the type of the attribute.

NOTE If at least one attribute has no read access right under the current association, then a read_by_logicalname () to attribute index 0 reveals the error message “scope-of-access-violated”.

5.4.3.3.4 Reserved from previous version

This method is reserved.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.3.3.5 change_secret(data)

Changes the LLS or HLS secret (for example password).

data::= octet-string new secret

NOTE 1 The structure of the “new secret” depends on the security mechanism implemented. The “new secret” may contain additional check bits and it may be encrypted.

NOTE 2 In the case of HLS with GMAC, the (HLS_) secret is held by the “Security setup“ object referenced in attribute

5.4.3.3.6 Reserved from previous version

This method is reserved.

5.4.3.3.7 Reserved from previous version

This method is reserved.

5.4.3.3.8 reply_to_HLS_authentication (data)

The remote invocation of this method delivers to the server the result of the secret processing by the client of the server’s challenge to the client, f(StoC), as the data service parameter of the Read.request primitive invoked with parameterised access.

data::= octet-string client’s response to the challenge

If the authentication is accepted, then the response (Read.confirm primitive) contains Result == OK and the result of the secret processing by the server of the client’s challenge to the server, f(CtoS) in the data service parameter of the Read.response service.

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

5.4.4 Association SN (Class_id = 12, version =3)

5.4.4.1 Overview

NOTE This new version 3 of the “Association SN” IC supports the client user identification process, see 4.4.2.

COSEM logical devices able to establish AAs within a COSEM context using SN referencing, model the AAs using instances of the “Association SN” IC. A COSEM logical device may have one instance of this IC for each AA the device is able to support.

The **short_name** of the “Association SN” object itself is fixed within the COSEM context. See 4.1.3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Association SN	0...n	class_id = 12, version = 3			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. object_list (static)	objlist_type				x + 0x08
3. access_rights_list (static)	access_rights_type				x + 0x10
4. security_setup_reference (static)	octet-string				x + 0x18
5. user_list (static)	array				x + 0x20
6. current_user	structure				x + 0x28
Specific methods	***m/o***				
1. *reserved from previous versions*	o				
2. *reserved from previous versions*	o				
3. read_by_logicalname (data)	o				x + 0x30
4. *reserved from previous versions*	o				
5. change_secret (data)	o				x + 0x40
6. *reserved from previous versions*	o				
7. *reserved from previous versions*					
8. reply_to_HLS_authentication (data)	o				x + 0x58
9. add_user (data)	o				x + 0x60
10. remove_user (data)	o				x + 0x68

5.4.4.2 Attribute description

5.4.4.2.1 logical_name

Identifies the “Association SN” object instance. See 6.2.33.

5.4.4.2.2 object_list

Contains the list of all objects with their base_name (short_name), class_id, version and *logical_name*. The base_name is the DLMS objectName of the first attribute (*logical_name*).

```
objlist_type::= array            objlist_element
objlist_element::= structure
{
        base_name:      long,
        class_id:       long-unsigned,
        version:        unsigned,
        logical_name:   octet-string
}
```

selective access (see 4.1.4) to the attribute object_list may be available. The access selector values and their parameters are as defined in 5.4.4.2.7.

5.4.4.2.3 access_rights_list

Contains the access rights to attributes and methods.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The link between the *object_list* and the *access_rights_list* is the base_name, present in both the objlist_element structure and the access_right_element structure. Therefore, the base_names on the two lists shall be the same. The number – and preferably, the order – of the elements in the array of objlist_element and the array of access_right_element shall also be the same.

```
access_rights_type::= array          access_rights_element

access_rights_element::= structure

{
      base_name:                     long,
      attribute_access:              attribute_access_descriptor,
      method_access:                 method_access_descriptor
}

attribute_access_descriptor::= array      attribute_access_item

attribute_access_item::= structure
{
      attribute_id:                  integer,
      access_mode:                   enum:
                  (0)   no_access,
                  (1)   read_only,
                  (2)   write_only,
                  (3)   read_and_write,
                  (4)   authenticated_read_only,
                  (5)   authenticated_write_only,
                  (6)   authenticated_read_and_write
      access_selectors:              CHOICE
      {
            null-data                [0],
            array integer            [1]
      }
}

method_access_descriptor::= array      method_access_item

method_access_item::= structure
{
      method_id:                     integer,
      access_mode:                   enum:
                                          (0)   no_access,
                                          (1)   access,
                                          (2)   authenticated_access
}
```

selective access (see 4.1.4) to the *attribute access_rights_list* may be available (optional). The access selector values and their parameters are as defined below.

5.4.4.2.4 security_setup_ reference

References the “Security setup” object by its logical_name. The referenced object manages security for a given “Association SN” object instance.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.4.2.5 user_list

Contains the list of users allowed to use the AA managed by the given instance of the "Association SN" IC.

```
array        user_list_entry

user_list_entry::= structure
{
        user_id:          unsigned,
        user_name:        visible-string
}
```

Where:

– user_id is the identifier of the user (this value is carried by the calling-AE-invocation-id field of the AARQ);

– user_name is the name of the user.

If the user_list attribute is empty – i.e. it is an array of 0 elements – any user can use the AA, i.e. the calling-AE-invocation-id field of the AARQ is ignored.

If the user_list attribute is not empty then only the users in the list can establish the AA, i.e. the calling-AE-invocation-id field of the AARQ shall be present and its value shall match one of user_ids in the user_list or else, the AA is not established.

5.4.4.2.6 current_user

Holds the identifier of the current user.

current_user::= user_list_entry (see 5.4.4.2.5).

If the user_list is empty, then current_user shall be a structure {user_id: unsigned 0, user_name: visible string of 0 elements}.

5.4.4.2.7 Parameters for selective access to the *object_list* attribute

Table 46 provides the parameters for selective access to *object_list* attribute.

Table 46 – Parameters for selective access to the object_list attribute

Access selector value	Parameter	Comment
1	class_id: long-unsigned	Delivers the subset of the object_list for a specific class_id. For the response: data::= objlist_type

2	structure
	{
	class_id: long-unsigned,
	logical_name: octet-string
	}
		Delivers the entry of the object_list for a specific class_id and logical_name.
		For the response:
		data::= objlist_element

5.4.4.3 Method description

5.4.4.3.1 Reserved from previous version

This method is reserved.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.4.3.2 Reserved from previous version

This method is reserved.

5.4.4.3.3 read_by_logicalname (data)

Reads attributes for selected objects. The objects are specified by their class_id and their logical_name. With this method, the parameterized access feature can also be used.

```
data::= array       attribute_identification

attribute_identification::= structure
{
        class_id:               long-unsigned,
        logical_name:           octet-string,
        attribute_index:        integer
}
```

Where:

– attribute_index is a pointer (i.e. offset) to the attribute within the object;

– attribute_index 0 delivers all attributes; attribute_index 1 delivers the first attribute (i.e. logical_name), etc.).

For the response: data is according to the type of the attribute.

NOTE If at least one attribute has no read access right under the current association, then a read_by_logicalname () to attribute index 0 reveals the error message “scope-of-access-violated”.

5.4.4.3.4 Reserved from previous version

This method is reserved.

5.4.4.3.5 change_secret (data)

Changes the LLS or HLS secret (for example password).

```
data::= octet-string       new secret
```

NOTE The structure of the “new secret” depends on the security mechanism implemented. The “new secret” may contain additional check bits and it may be encrypted.

5.4.4.3.6 Reserved from previous version

This method is reserved.

5.4.4.3.7 Reserved from previous version

This method is reserved.

5.4.4.3.8 reply_to_HLS_authentication (data)

The remote invocation of this method delivers to the server the result of the secret processing by the client of the server’s challenge to the client, f(StoC), as the data service parameter of the Read.request primitive invoked with parameterised access.

```
data::= octet-string       client’s response to the challenge
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If the authentication is accepted, then the response (Read.confirm primitive) contains Result == OK and the result of the secret processing by the server of the client's challenge to the server, f(CtoS) in the data service parameter of the Read.response service.

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

5.4.4.3.9 add_user (data)

Adds a user to the user_list.

data::= user_list_entry (see 5.4.4.2.5).

5.4.4.3.10 remove_user (data)

Removes a user from the user_list.

data::= user_list_entry (see 5.4.4.2.5).

5.4.5 Association LN (class_id = 15, version = 0)

5.4.5.1 Overview

This IC allows modelling of AAs between a server and a client, using the application context *logical name referencing*. Each AA is modelled by one Association LN object.

Association LN	**0...MaxNbofAss.**	**class_id = 15, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. object_list (static)	object_list_type				x + 0x08
3. associated_partners_id	associated_partners_type				x + 0x10
4. application_context_name	context_name_type				x + 0x18
5. xDLMS_context_info	xDLMS-context-type				x + 0x20
6. authentication_mechanism_name	mechanism_name_type				x + 0x28
7. LLS_secret	octet-string				x + 0x30
8. association_status	enum				x + 0x38
Specific methods	***m/o***				
1. reply_to_HLS_authentication (data)	o				x + 0x60
2. change_HLS_secret (data)	o				x + 0x68
3. add_object (data)	o				x + 0x70
4. remove_object (data)	o				x + 0x78

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.5.2 Attribute description

5.4.5.2.1 logical_name

Identifies the "Association LN" object instance. See 6.2.33.

5.4.5.2.2 object_list

Contains the list of visible COSEM objects with their class_id, version, logical name and the access rights to their attributes and methods within the given application association.

```
object_list_type::= array object_list_element
object_list_element::= structure
{
        class_id:               long-unsigned,
        version:                unsigned,
        logical_name:           octet-string,
        access_rights:          access_right
}

access_right::= structure
{
        attribute_access:       attribute_access_descriptor,
        method_access:          method_access_descriptor
}

attribute_access_descriptor::= array attribute_access_item
attribute_access_item::= structure
{
        attribute_id:       integer,
        access_mode:        enum:
                                (0)   no_access,
                                (1)   read_only,
                                (2)   write_only,
                                (3)   read_and_write
        access_selectors:   CHOICE
                {
                    null-data           [0],
                    array integer       [1]
                }
}

method_access_descriptor::= array  method_access_item
method_access_item::=  structure
{
  method_id: integer,
  access_mode:     boolean
}
```

Where:

– the attribute_access_descriptor and the method_access_ descriptor always contain all implemented attributes or methods;

– access_selectors contain a list of the supported selector values;

– selective access (see 4.1.4) to the attribute object_list may be available (optional).

The selective access parameters are as defined in 5.4.5.2.9.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.5.2.3 associated_partners_id

Contains the identifiers of the COSEM client and server (logical device) APs within the physical devices hosting these processes, which belong to the AA modelled by the "Association LN" object.

```
associated_partners_type::= structure
{
    client_SAP:     integer,
    server_SAP:     long-unsigned
}
```

The range for the client_SAP is 0…0x7F.

The range for the server_SAP is 0x000…0x3FFF.

5.4.5.2.4 application_context_name

In the COSEM environment, it is intended that an application context pre-exists and is referenced by its name during the establishment of an AA. This attribute contains the name of the application context for that AA.

```
context_name_type::=        CHOICE
{
    context_name_structure      [2],
    octet-string                [9]
}
}
```

The application context name is specified as OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.2.

When the context_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
context_name_structure::= structure

{
    joint_iso_ctt_element:              unsigned,
    country_element:                    unsigned,
    country_name_element:               long-unsigned,
    identified_organization_element:    unsigned,
    DLMS_UA_element:                    unsigned,
    application_context_element:        unsigned,
    context_id_element:                 unsigned
}
```

EXAMPLE 1: In the case of context_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 01 11 01 (all values are hexadecimal).

When the context_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

EXAMPLE 2: In the case of context_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 01 01 (all values are hexadecimal).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.5.2.5 xDLMS_context_info

Contains all the necessary information on the xDLMS context for the given AA.

```
xDLMS-context-type::= structure
{
    conformance:                bit-string,
    max_receive_pdu_size:       long-unsigned,
    max_send_pdu_size:          long-unsigned,
    dlms_version_number:        unsigned,
    quality_of_service:         integer,
    cyphering_info:             octet-string
}
```

Where:

– the conformance element contains the xDLMS conformance block supported by the server;

– the max_receive_pdu_size element contains the maximum length for an xDLMS APDU, expressed in bytes that the client may send. This is the same as the server-max-receive-pdu-size parameter of the xDLMS initiateResponse APDU (see IEC 62056-5-3:2023, Clause 8);

– the max_send_pdu_size, in an active association contains the maximum length for an xDLMS APDU, expressed in bytes that the server may send. This is the same as the client-max-receive-pdu-size parameter of the xDLMS initiateRequest APDU (see IEC 62056-5-3:2023, Clause 8);

– the dlms_version_number element contains the DLMS version number supported by the server;

– the quality_of_service element is not used;

– the cyphering_info, in an active association, contains the dedicated key parameter of the xDLMS initiateRequest APDU (see IEC 62056-5-3:2023, Clause 8).

5.4.5.2.6 authentication_ mechanism_name

Contains the name of the authentication mechanism for the AA.

```
mechanism_name_type::=    CHOICE
{
    mechanism_name_structure        [2],
    octet-string                    [9]
}
```

The authentication mechanism name is specified as an OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.3.

When the mechanism_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
mechanism_name_structure::= structure
{
    joint_iso_ctt_element:                  unsigned,
    country_element:                        unsigned,
    country_name_element:                   long-unsigned,
    identified_organization_element:        unsigned,
    DLMS_UA_element:                        unsigned,
    authentication_mechanism_name_element:  unsigned,
    mechanism_id_element:                   unsigned
}
```

EXAMPLE 1: In the case of mechanism_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 02 11 01 (all values are hexadecimal).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

When the mechanism_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

EXAMPLE 2: In the case of mechanism_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 02 01 (all values are hexadecimal).

No mechanism_name is required when no authentication is used.

5.4.5.2.7 LLS_secret

Contains the authentication value for the LLS authentication process.

5.4.5.2.8 association_status

Indicates the current status of the association, which is modelled by the object.

enum:

(0) non-associated,
(1) association-pending,
(2) associated

5.4.5.2.9 Parameters for selective access to the *object_list* attribute

- if no selective access is requested, (no Access_Selection_Parameters parameter is present in the GET.request (.indication) service primitive for the object_list attribute) the corresponding .response (.confirmation) service shall contain all object_list_element of the *object_list* attribute;
- when selective access is requested to the *object_list* attribute (the Access_Selection_Parameters parameter is present), the response shall contain a 'filtered' list of object_list_element, as shown in Table 47.

Table 47 – Parameters for selective access to the *object_list* attribute

Access selector	Access parameter	Comment
1	NULL	All information excluding the access_rights shall be included in the response.
2	class_list	Access by class. In this case, only those object_list_element of the *object_list* shall be included in the response, which have a class_id equal to one of the class_id-s of the class-list. No access_right information is included. class_list::= array class_id class_id::= long-unsigned

3	object_id_list	Access by object. The full information record of object instances on the object_id_list shall be returned.
		object_id_list ::= array object_id
		object_id::= structure
		{
		class_id: long-unsigned,
		logical_name: octet-string
		}

4	object_id	The full information record of the required COSEM object instance shall be returned. object_id: See above.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.5.3 Method description

5.4.5.3.1 reply_to_HLS_authentication (data)

The remote invocation of this method delivers the client's “secretly” processed “challenge StoC” (f(StoC)) back to the server as the *data* service parameter of the ACTION.request primitive invoked.

data::= octet-string client’s response to the challenge

If the authentication is accepted, then the response (ACTION.confirm primitive) contains Result == OK and the server’s "secretly" processed "challenge CtoS" (f(CtoS)) back to the client in the *data* service parameter of the response service.

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

5.4.5.3.2 change_HLS_secret (data)

Changes the HLS secret (for example encryption key).

data::= octet-string[a] new HLS secret

NOTE The structure of the “new secret” depends on the security mechanism implemented. The “new secret” may contain additional check bits and it may be encrypted.

5.4.5.3.3 add_object (data)

Adds the referenced object to the object_list.

data::= object_list_element (see 5.4.5.2.2).

5.4.5.3.4 remove_object (data)

Removes the referenced object from the object_list.

data::= object_list_element (see 5.4.5.2.2).

5.4.6 Association LN (class_id = 15, version = 1)

5.4.6.1 Overview

COSEM logical devices able to establish AAs within a COSEM context using LN referencing, model the AAs through instances of the “Association LN” IC. A COSEM logical device has one instance of this IC for each AA the device is able to support.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Association LN		0...MaxNbofAss.	class_id = 15, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. object_list	(static)	object_list_type				x + 0x08
3. associated_partners_id		associated_partners_type				x + 0x10
4. application_context_name		context_name_type				x + 0x18
5. xDLMS_context_info		xDLMS_context_type				x + 0x20
6. authentication_mechanism_name		mechanism_name_type				x + 0x28
7. secret		octet-string				x + 0x30
8. association_status		enum				x + 0x38
9. security_setup_reference	(static)	octet-string				x + 0x40
Specific methods		***m/o***				
1. reply_to_HLS_authentication (data)		o				x + 0x60
2. change_HLS_secret (data)		o				x + 0x68
3. add_object (data)		o				x + 0x70
4. remove_object (data)		o				x + 0x78

5.4.6.2 Attribute description

5.4.6.2.1 logical_name

Identifies the “Association LN” object instance. See 6.2.33.

5.4.6.2.2 object_list

Contains the list of visible COSEM objects with their class_id, version, logical name and the access rights to their attributes and methods within the given application association.

```
object_list_type::= array      object_list_element

object_list_element::= structure
{
        class_id:              long-unsigned,
        version:               unsigned,
        logical_name:          octet-string,
        access_rights:         access_right
}

access_right::= structure
{
        attribute_access:      attribute_access_descriptor,
        method_access:         method_access_descriptor
}

attribute_access_descriptor::= array      attribute_access_item
attribute_access_item::=       structure
{
        attribute_id:          integer,
                access_mode:   enum:
                                   (0)   no_access,
                                   (1)   read_only,
                                   (2)   write_only,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
                                        (3)    read_and_write,
                                        (4)    authenticated_read_only,
                                        (5)    authenticated_write_only,
                                        (6)
                                        authenticated_read_and_write
        access_selectors:       CHOICE
            {
                null-data           [0],
                array integer       [1]
            }
}
method_access_descriptor::= arraymethod_access_item

method_access_item::= structure
{
    method_id: integer,
    access_mode:    enum:
                            (0)    no_access,
                            (1)    access,
                            (2)    authenticated_access
}
```

Where:

– the attribute_access_descriptor and the method_access_descriptor elements always contain all implemented attributes or methods;

– access_selectors contain a list of the supported selector values.

selective access (see 4.1.4) to the attribute object_list may be available (optional). The selective access parameters are as defined in 5.4.5.2.9.

5.4.6.2.3 associated_partners_ id

Contains the identifiers of the COSEM client and server (logical device) APs within the physical devices hosting these APs, which belong to the AA modelled by the “Association LN” object.

```
associated_partners_type::= structure
{
    client_SAP:integer,
    server_SAP:     long-unsigned
}
```

The range for the client_SAP is 0…0x7F.

The range for the server_SAP is 0x0000…0x3FFF.

The SAPs shall be in the range allowed by the data type and the media.

5.4.6.2.4 application_context_name

In the COSEM environment, it is intended that an application context pre-exists and is referenced by its name during the establishment of an AA. This attribute contains the name of the application context for that AA.

```
context_name_type::=        CHOICE
{
    context_name_structure      [2],
    octet-string                [9]
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The application context name is specified as OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.2.

When the context_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
context_name_structure::= structure
{
        joint_iso_ctt_element:              unsigned,
        country_element:                    unsigned,
        country_name_element:               long-unsigned,
        identified_organization_element:    unsigned,
        DLMS_UA_element:                    unsigned,
        application_context_element:        unsigned,
        context_id_element:                 unsigned
}
```

Example 1: In the case of context_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 01 11 01 (all values are hexadecimal).

When the context_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

Example 2: In the case of context_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 01 01 (all values are hexadecimal).

5.4.6.2.5 xDLMS_context_info

Contains all the necessary information on the xDLMS context for the given AA.

```
xDLMS-context-type::= structure
{
        conformance:                bit-string,
        max_receive_pdu_size:       long-unsigned,
        max_send_pdu_size:          long-unsigned,
        dlms_version_number:        unsigned,
        quality_of_service:         integer,
        cyphering_info:             octet-string
}
```

Where:

– the conformance element contains the xDLMS conformance block supported by the server;

– the max_receive_pdu_size element contains the maximum length for an xDLMS APDU, expressed in bytes that the client may send. This is the same as the server-max-receive-pdu-size parameter of the xDLMS initiateResponse APDU (see IEC 62056-5-3:2023, Clause 8);

– the max_send_pdu_size, in an active association contains the maximum length for an xDLMS APDU, expressed in bytes that the server may send. This is the same as the client-max-receive-pdu-size parameter of the xDLMS initiateRequest APDU (see IEC 62056-5-3:2023, Clause 8);

– the dlms_version_number element contains the DLMS® version number supported by the server;

– the quality_of_service element is not used;

– the cyphering_info, in an active association, contains the dedicated key parameter of the xDLMS initiateRequest APDU (see IEC 62056-5-3:2023, Clause 8).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.6.2.6 authentication_mechanism_name

Contains the name of the authentication mechanism for the AA.

```
mechanism_name_type::=    CHOICE
{
        mechanism_name_structure        [2],
        octet-string                    [9]
}
```

The authentication mechanism name is specified as an OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.3.

When the mechanism_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
mechanism_name_structure::= structure
{
        joint_iso_ctt_element:                  unsigned,
        country_element:                        unsigned,
        country_name_element:                   long-unsigned,
        identified_organization_element:        unsigned,
        DLMS_UA_element:                        unsigned,
        authentication_mechanism_name_element:  unsigned,
        mechanism_id_element:                   unsigned
}
```

Example 1: In the case of mechanism_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 02 11 01 (all values are hexadecimal).

When the mechanism_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

EXAMPLE 2: In the case of mechanism_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 02 01 (all values are hexadecimal).

No mechanism_name is required when no authentication is used.

5.4.6.2.7 secret

Contains the secret for the LLS or HLS authentication process.

NOTE In the case of HLS with GMAC, the (HLS_)secret is held by the Security setup object referenced in attribute 9.

5.4.6.2.8 association_status

Indicates the current status of the association, which is modelled by the object.

```
enum:

(0)  non-associated,
(1)  association-pending,
(2)  associated.
```

5.4.6.2.9 security_setup_ reference

References the Security setup object by its logical name. The referenced object manages security for a given Association LN object instance.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

A SET operation on an attribute of an association LN object becomes effective when this association object is used to establish a new association.

5.4.6.3 Method description

5.4.6.3.1 reply_to_HLS_authentication (data)

The remote invocation of this method delivers to the server the result of the secret processing by the client of the server's challenge to the client, f(StoC), as the *data* service parameter of the ACTION.request primitive invoked.

data::= octet-string client's response to the challenge

If the authentication is accepted, then the response (ACTION.confirm primitive) contains Result == OK and the result of the secret processing by the server of the client's challenge to the server, f(CtoS) in the *data* service parameter of the response service.

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

5.4.6.3.2 change_HLS_secret (data)

Changes the HLS secret (for example encryption key).

data::= octet-string new HLS secret

The structure of the "new secret" depends on the security mechanism implemented. The "new secret" may contain additional check bits and it may be encrypted.

5.4.6.3.3 add_object (data)

Adds the referenced object to the object_list.

data::= object_list_element

5.4.6.3.4 remove_object (data)

Removes the referenced object from the object_list.

data::= object_list_element .

5.4.7 Association LN (class_id = 15, version = 2)

5.4.7.1 Overview

NOTE This version 2 of the "Association LN" IC supports the client user identification process, see 4.4.2.

COSEM logical devices able to establish AAs within a COSEM context using LN referencing, model the AAs through instances of the "Association LN" IC. A COSEM logical device has one instance of this IC for each AA the device is able to support.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Association LN		0...MaxNbofAss.	class_id = 15, version = 2			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. object_list	(static)	object_list_type				x + 0x08
3. associated_partners_id		associated_partners				x + 0x10
4. application_context_name		context_name_type				x + 0x18
5. xDLMS_context_info		xDLMS_context_type				x + 0x20
6. authentication_0mechanism_name		mechanism_name_type				x + 0x28
7. secret		octet-string				x + 0x30
8. association_status		enum				x + 0x38
9. security_setup_reference	(static)	octet-string				x + 0x40
10. user_list	(static)	array				x + 0x48
11. current_user		structure				x + 0x50
Specific methods		***m/o***				
1. reply_to_HLS_authentication (data)		o				x + 0x60
2. change_HLS_secret (data)		o				x + 0x68
3. add_object (data)		o				x + 0x70
4. remove_object (data)		o				x + 0x78
5. add_user (data)		o				x + 0x80
6. remove_user (data)		o				x + 0x88

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.7.2 Attribute description

5.4.7.2.1 logical_name

Identifies the “Association LN” object instance. See 6.2.33.

5.4.7.2.2 object_list

Contains the list of visible COSEM objects with their class_id, version, *logical_ name* and the access rights to their attributes and methods within the given AA.

```
object_list_type::= array       object_list_element

object_list_element::= structure
{
        class_id:               long-unsigned,
        version:                unsigned,
        logical_name:           octet-string,
        access_rights:          access_right
}
access_right::= structure
{
        attribute_access:       attribute_access_descriptor,
        method_access:          method_access_descriptor
}

attribute_access_descriptor::= array        attribute_access_item

attribute_access_item::=        structure
```

```
{
    attribute_id:       integer,
    access_mode:        enum:
                            (0)   no_access,
                            (1)   read_only,
                            (2)   write_only,
                            (3)   read_and_write,
                            (4)   authenticated_read_only,
                            (5)   authenticated_write_only,
                            (6)   authenticated_read_and_write

    access_selectors:       CHOICE
    {
        null-data           [0],
        array integer       [1]
    }
}

method_access_descriptor::= arraymethod_access_item

method_access_item::= structure
{
    method_id:          integer,
    access_mode:        enum:
                            (0)   no_access,
                            (1)   access,
                            (2)   authenticated_access
}
```

Where:

– the attribute_access_descriptor and the method_access_descriptor elements always contain all implemented attributes or methods;

– access_selectors contain a list of the supported selector values.

Selective access (see 4.1.4) to the attribute object_list may be available (optional). The selective access parameters are as defined in 5.4.5.2.9.

5.4.7.2.3 associated_partners_ id

Contains the identifiers of the COSEM client and server (logical device) APs within the physical devices hosting these APs, which belong to the AA modelled by the “Association LN” object.

```
associated_partners_type::= structure
{
    client_SAP:integer,
    server_SAP:     long-unsigned
}
```

The range for the client_SAP is 0…0x7F.

The range for the server_SAP is 0x0000…0x3FFF.

The SAPs shall be in the range allowed by the data type and the media.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.7.2.4 application_context_name

In the COSEM environment, it is intended that an application context pre-exists and is referenced by its name during the establishment of an AA. This attribute contains the name of the application context for that AA.

```
context_name_type::=        CHOICE
{
        context_name_structure      [2],
        octet-string                [9]
}
```

The application context name is specified as OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.2.

When the context_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
context_name_structure::= structure

{
        joint_iso_ctt_element:              unsigned,
        country_element:                    unsigned,
        country_name_element:               long-unsigned,
        identified_organization_element:    unsigned,
        DLMS_UA_element:                    unsigned,
        application_context_element:        unsigned,
        context_id_element:                 unsigned
}
```

Example 1: In the case of context_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 01 11 01 (all values are hexadecimal).

When the context_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

Example 2: In the case of context_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 01 01 (all values are hexadecimal).

5.4.7.2.5 xDLMS_context_info

Contains all the necessary information on the xDLMS context for the given AA.

```
xDLMS-context-type::= structure
{
        conformance:                bit-string,
        max_receive_pdu_size:       long-unsigned,
        max_send_pdu_size:          long-unsigned,
        dlms_version_number:        unsigned,
        quality_of_service:         integer,
        cyphering_info:             octet-string
}
```

Where:

– the conformance element contains the xDLMS conformance block supported by the server;

– the max_receive_pdu_size element contains the maximum length for an xDLMS APDU, expressed in bytes that the client may send. This is the same as the server-max-receive-pdu-size parameter of the xDLMS initiateResponse APDU (see IEC 62056-5-3:2023, Clause 8);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– the max_send_pdu_size, in an active association contains the maximum length for an xDLMS APDU, expressed in bytes that the server may send. This is the same as the client-max-receive-pdu-size parameter of the xDLMS initiateRequest APDU (see IEC 62056-5-3:2023, Clause 8);
– the dlms_version_number element contains the DLMS® version number supported by the server;
– the quality_of_service element is not used;
– the cyphering_info, in an active association, contains the dedicated key parameter of the xDLMS initiateRequest APDU (see IEC 62056-5-3:2023, Clause 8).

5.4.7.2.6 authentication_mechanism_name

Contains the name of the authentication mechanism for the AA.

```
mechanism_name_type::=    CHOICE
{
        mechanism_name_structure        [2],
        octet-string                    [9]
}
```

The authentication mechanism name is specified as an OBJECT IDENTIFIER in IEC 62056-5-3:2023, 7.2.2.3.

When the mechanism_name_type is encoded as a structure, it includes the arc labels of the OBJECT IDENTIFIER.

```
mechanism_name_structure::= structure
{
        joint_iso_ctt_element:                  unsigned,
        country_element:                        unsigned,
        country_name_element:                   long-unsigned,
        identified_organization_element:        unsigned,
        DLMS_UA_element:                        unsigned,
        authentication_mechanism_name_element:  unsigned,
        mechanism_id_element:                   unsigned
}
```

Example 1: In the case of mechanism_id(1) the A-XDR encoding is: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 02 11 01 (all values are hexadecimal).

When the mechanism_name_type is encoded as an octet-string, it holds the value of the OBJECT IDENTIFIER. See IEC 62056-5-3:2023, Clause D.4.

EXAMPLE 2: In the case of mechanism_id(1) the A-XDR encoding is: 09 07 60 85 74 05 08 02 01 (all values are hexadecimal).

No mechanism_name is required when no authentication is used.

5.4.7.2.7 secret

Contains the secret for the LLS or HLS authentication process.

NOTE In the case of HLS with GMAC, the (HLS_)secret is held by the Security setup object referenced in attribute 9.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.7.2.8 association_status

Indicates the current status of the association, which is modelled by the object.

enum:

(0) non-associated,
(1) association-pending,
(2) associated.

5.4.7.2.9 security_setup_ reference

References the Security setup object by its logical name. The referenced object manages security for a given Association LN object instance.

A SET operation on an attribute of an association LN object becomes effective when this association object is used to establish a new association.

5.4.7.2.10 user_list

Contains the list of users allowed to use the AA managed by the given instance of the "Association LN" IC.

```
array        user_list_entry
user_list_entry::= structure
{
        user_id:     unsigned,
        user_name:       visible-string
}
```

Where:

– user_id is the identifier of the user (this value is carried by the calling-AE-invocation-id field of the AARQ);
– user_name is the name of the user.

If the *user_list* attribute is empty – i.e. it is an array of 0 elements – any user can use the AA, i.e. the calling-AE-invocation-id field of the AARQ is ignored.

If the *user_list* attribute is not empty then only the users in the list can establish the AA, i.e. the calling-AE-invocation-id field of the AARQ shall be present and its value shall match one of user_ids in the *user_list* or else the AA is not established.

5.4.7.2.11 current_user

Holds the identifier of the current user.

current_user::= user_list_entry (see 5.4.7.2.10).

If the user_list is empty, then current_user shall be a structure {user_id: unsigned 0, user_name: visible string of 0 elements}

5.4.7.3 Method description

5.4.7.3.1 reply_to_HLS_authentication (data)

The remote invocation of this method delivers to the server the result of the secret processing by the client of the server's challenge to the client, f(StoC), as the *data* service parameter of the ACTION.request primitive invoked.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

data::= octet-string client's response to the challenge

If the authentication is accepted, then the response (ACTION.confirm primitive) contains Result == OK and the result of the secret processing by the server of the client's challenge to the server, f(CtoS) in the *data* service parameter of the response service.

data::= octet-string server's response to the challenge

If the authentication is not accepted, then the result parameter in the response shall contain a non-OK value, and no data shall be sent back.

5.4.7.3.2 change_HLS_secret (data)

Changes the HLS secret (for example encryption key).

data::= octet-string new HLS secret

The structure of the "new secret" depends on the security mechanism implemented. The "new secret" may contain additional check bits and it may be encrypted.

5.4.7.3.3 add_object (data)

Adds the referenced object to the object_list.

data::= object_list_element

5.4.7.3.4 remove_object (data)

Removes the referenced object from the object_list.

data::= object_list_element .

5.4.7.3.5 add_user (data)

Adds a user to the *user_list*.

data::= user_list_entry (see 5.4.7.2.10).

5.4.7.3.6 remove_user (data)

Removes a user from the *user_list*.

data::= user_list_entry (see 5.4.7.2.10).

5.4.8 Security setup (class_id = 64, version = 0)

5.4.8.1 Overview

Instances of this IC contain the necessary information on the security policy applicable and the security suite in use within a particular AA, between two systems identified by their client system title and server system title respectively. They also contain methods to increase the level of security and to transfer the global keys. See IEC 62056-5-3:2023, Clause 5.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Security setup		0...n	class_id = 64, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. security_policy	(static)	enum	0	3	0	x + 0x08
3. security_suite	(static)	enum	0	0	0	x + 0x10
4. client_system_title	(dyn.)	octet-string				x + 0x18
5. server_system_title	(static)	octet-string				x + 0x20
Specific methods		***m/o***				
1. security_activate (data)		o				x + 0x28
2. global_key_transfer (data)		o				x + 0x30

5.4.8.2 Attribute description

5.4.8.2.1 logical_name

Identifies the “Security setup” object instance. See 6.2.36.

5.4.8.2.2 security_policy

Enforces authentication and/or encryption algorithm provided with security_suite.

enum:

(0) nothing,
(1) all messages to be authenticated,
(2) all messages to be encrypted,
(3) all messages to be authenticated and encrypted
(4) ...(15) reserved

5.4.8.2.3 security_suite

Specifies authentication, encryption and key transport algorithm.

enum:

(0) AES-GCM-128 for authenticated encryption and AES-128 for key wrapping
(1) ...(15) reserved

5.4.8.2.4 client_system_title

Carries the (current) client system title:

– in the S-FSK PLC environment, the active initiator sends its system title using the CIASE protocol;

NOTE It is also held by the *active_initiator* attribute of the S-FSK Active initiator object; see 4.10.4;

– during confirmed or unconfirmed AA establishment, it is carried by the calling-AP-title field of the AARQ APDU;

– If a client system title has already been sent during a registration process, like in the case of the S-FSK PLC profile, the client system title carried by the AARQ APDU should be the same. If not, the AA shall be rejected and appropriate diagnostic information shall be sent;

– in a pre-established AA, it can be written by the client using an unsecured SET / Write service.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.8.2.5 server_system_title

Carries the server system title.

– in the S-FSK PLC environment, the server sends its system title during the discover process, using the CIASE protocol;
– during confirmed AA establishment, it is carried by the responding-AP-title field of the AARE APDU.

This attribute shall be read only.

5.4.8.3 Method description

5.4.8.3.1 security_activate (data)

Activates and strengthens the security policy:

```
enum:

(0)  nothing,
(1)  all messages to be authenticated,
(2)  all messages to be encrypted,
(3)  all messages to be authenticated and encrypted
```

The new security policy applies as soon as the method invocation has been confirmed with success.

NOTE The security policy can only be strengthened.

5.4.8.3.2 global_key_transfer (data)

Updates one or more global keys. The *data* parameter includes wrapped key data. Key data include the key identifiers and the keys themselves.

```
array       key_data

key_data::= structure
{
      key_id:      enum:
                        (0) global unicast encryption key,
                        (1) global broadcast encryption key,
                        (2) authentication key
      key_wrapped:     octet-string
}
```

The key wrapping algorithm is as specified by the security suite. The KEK is the master key.

The new key(s) are valid as soon as the method invocation has been confirmed with success.

5.4.9 Push Setup (class_id = 40, version = 0)

5.4.9.1 Overview

The "Push setup" IC contains a list of references to COSEM object attributes to be pushed. It also contains the push destination and method as well as the communication time windows and the handling of retries.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The push takes place upon invoking the *push* method, triggered by a Push "Single action schedule" object, by an alarm "Register monitor" object, by a dedicated internal event or externally. After the push operation has been triggered, it is executed according to the settings made in the given "Push setup" object. Depending on the communication window settings, the push is executed immediately or as soon as a communication window becomes active, after a random delay. If the push was not successful, retries are made. Push windows, delays and retries are shown in Figure 30.

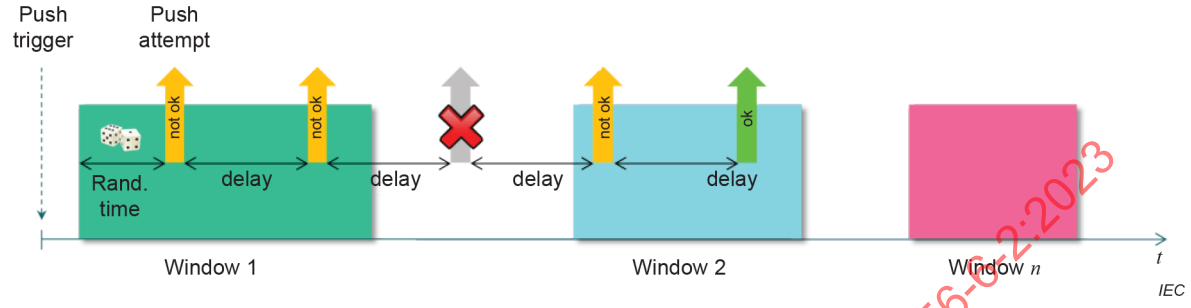


Figure 30 – Push windows and delays

Push setup		0...n	class_id = 40, version = 0			
Attribute (s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. push_object_list	(static)	array				x + 0x08
3. send_destination_and_method	(static)	structure				x + 0x10
4. communication_window	(static)	array				x + 0x18
5. randomisation_start_interval	(static)	long-unsigned				x + 0x20
6. number_of_retries	(static)	unsigned				x + 0x28
7. repetition_delay	(static)	long-unsigned				x + 0x30
Specific methods		***m/o***				
1. push (data)		m				x + 0x38

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.9.2 Attribute description

5.4.9.2.1 logical_name

Identifies the "Push setup" object instance. See 6.2.24.

5.4.9.2.2 push_object_list

Defines the list of attributes to be pushed.

Upon invocation of the *push* (data) method the items are sent to the destination defined in the *send_destination_and_method* attribute.

```
array        object_definition

object_definition::=      structure

{
      class_id:              long-unsigned,
      logical_name:          octet-string,
      attribute_index:       integer,
      data_index:            long-unsigned
}
```

Where:

– attribute_index is a pointer to the attribute within the object, identified by class_id and logical_name: attribute_index 1 refers to the 1st attribute (i.e. the logical_name), attribute_index 2 to the 2nd attribute etc.; attribute_index 0 refers to all public attributes;

– data_index is a pointer selecting one or several specific elements of an attribute with a complex data type (structure or array).

data_index	MS-Byte		LS-Byte
	Upper nibble	Lower nibble	

If the data_type of the attribute is simple, then data_index has no meaning.

If the attribute is a structure or an array, then data_index points to an element in the structure or array. The first element in the complex attribute is identified by data_index 1.

When the attribute is the *buffer* of a “Profile generic” object, the data_index carries selective access parameters.

– 0x0000 = identifies the whole attribute;

– 0x0001 to 0x0FFF = identifies one element in the complex attribute;

– 0x1000 to 0xFFFF = selective access to the array holding the buffer of a “Profile generic” object. The data-index selects entries within a number of last (recent) time periods, or a number of last (recent) entries, as well as the columns in the array.

The encoding is specified in Table 48.

NOTE 1 If the push_object_list array is empty, the push operation is disabled.

NOTE 2 The push_object_list attribute itself can be pushed as well to clearly identify the data pushed.

Similarly to the case of the “Profile generic” IC, all attributes included in the push_object_list attribute are pushed regardless of the access rights to them. Therefore, writing of the push_object_list attribute should be restricted to clients with appropriate access rights.

5.4.9.2.3 send_destination_and_method

Contains the destination address (e.g. phone number, email address, IP address) where the data specified by the push_object_list has to be sent, as well as the sending method.

```
send_destination_and_method::= structure

{
      transport_service:     transport_service_type,
      destination:           octet-string,
      message:               message_type
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

– the transport_service element defines the type of service used to push the data:

transport_service_type::= enum:

(0) TCP,
(1) UDP,
(2) reserved for FTP,
(3) reserved for SMTP,
(4) SMS,
(5) HDLC,
(6) reserved for M-Bus,
(7) reserved for ZigBee®,
(200...255) manufacturer specific

– the destination element contains the target address where the data has to be sent. The elements of the target address depend on the transport service used.

Each “Push setup” object instance specifies a single destination. If it is required to push data to several destinations, several “Push setup” objects have to be instantiated,

– the message_type element identifies the encoding of the xDLMS APDU used.

message_type::= enum:

(0) A-XDR encoded xDLMS APDU,
(1) XML encoded xDLMS APDU,
(128...255) manufacturer specific

– all other transport_service_type and message_type values are reserved for future use.

5.4.9.2.4 communication_window

Defines the time points when the communication window(s) for the push become(s) active (start_time) and inactive (end_time). See Figure 30.

```
array       window_element

window_element::= structure

{
      start_time:  octet-string,
      end_time:    octet-string
}
```

start_time and end_time are formatted as specified in 4.1.6.1 for *date-time* including wildcards.

If the end of a communication window is reached an already started push operation is completed.

If no communication windows are defined (array [0]) the push operation is always possible.

5.4.9.2.5 randomisation_start_interval

To avoid simultaneous push operations by a lot of devices at exactly the same point in time, a randomisation interval – in seconds – can be defined. This means that the push operation is not started immediately after the invocation of the *push* method but randomly delayed within the interval.

The *randomisation_start_interval* attribute defines the maximum value which can be obtained from a randomizing algorithm. The resulting value is used to delay the first push operation. It is not used anymore in case of retries.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If no communication window is active when invoking the *push* method, the delay starts at the beginning of the next communication window. If the *push* method is invoked during an active communication window, the push operation is still delayed.

The *randomisation_start_interval* is only active for the first push attempt. If no *communication_window* is defined, the *randomisation_start_interval* is active for every push attempt.

If the *randomisation_start_interval* is set to 0 no delay is active.

If the randomisation time is longer than the first communication window, the first push attempt will be made during any later window.

5.4.9.2.6 number_of_retries

Defines the maximum number of retries in the case of unsuccessful or skipped push attempts. After a successful push operation no further push attempts are made until the push operation is triggered again. A push is treated as successful if a lower layer transmission confirmation has been received.

The conditions of detecting an unsuccessful push attempt may depend on the communication profile used and on the implementation and it is therefore out of the Scope of this document.

5.4.9.2.7 repetition_delay

The time delay, expressed in seconds until the next push attempt is started after an unsuccessful push.

NOTE 1 The repetition delay itself is not influenced by the communication window. But a retry only can be made if a communication window is active at that time. Otherwise it is handled like an unsuccessful push attempt.

NOTE 2 The push data is not stored in an intermediate buffer. In the case of retries, the current values of the attributes may change with every push attempt.

5.4.9.3 Method description

5.4.9.3.1 push (data)

Activates the push process leading to the elaboration and the sending of the push data taking into account the values of the attributes defined in the given instance of this IC.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 48 – Encoding of selective access parameters with data_index

	MS_Byte upper nibble: Selects the time periods or entries.
0xF	**Last complete number of months:** Selective access to the *buffer* resulting in all entries of the last complete number of months and the first entry at midnight of the current month.
0xE	**Last complete number of days:** Selective access to the *buffer* resulting in all entries of the last complete number of days and the first entry at midnight of today.
0xD	**Last complete number of hours:** Selective access to the *buffer* resulting in all entries of the last complete number of hours and the first entry of the current hour.
0xC	**Last complete number of minutes:** Selective access to the *buffer* resulting in all entries of the last complete number of minutes and the first entry of the current minute.
0xB	**Last number of seconds:** Selective access to the *buffer* resulting in all entries of the last number of seconds.
0xA	**Last complete number of months including the current month:** Same as 0xF above but all entries up to now are retrieved.
0x9	**Last complete number of days including the current day:** Same as 0xE above but all entries up to now are retrieved.
0x8	**Last complete number of hours including the current hour:** Same as 0xD above but all entries up to now are retrieved.
0x7	**Last complete number of minutes including the current minute:** This is the same as 0xC above but all entries up to now are retrieved.
0x6…0x2	Reserved
0x1	**Last number of entries**
0x0	**Whole attribute** or a single element in an attribute with complex data type is selected (MS-Byte lower nibble 0x0…0xF, LS-Byte 0x00…0xFF).
	MS-Byte lower nibble: Defines the number of columns of a “Profile generic” *buffer* selected.
0x0	All columns
0x1 to 0xF	Number of columns, starting from column 1.
0x00…0xFF	**LS-Byte:** – in the case of selective access by time period, (i.e. number of month, days, hours …) defines the number of recent complete time periods (1 to 255), – in the case of selective access by entry defines the number of recent entries.

Example 1)	0xE401: The entries for the last complete day are selected. The first 4 columns are included.
Example 2)	0xA300: The entries for zero last complete months and the current month are selected . The first 3 columns are included.
Example 3)	0x800C: The entries for the last complete 12 hours are selected. All columns are included.
Example 4)	0x1080: The last 128 entries are selected. All columns are included.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.4.10 Push Setup (class_id = 40, version = 1)

5.4.10.1 Overview

The "Push setup” interface class contains a list of references to COSEM object attributes to be pushed. It also contains the push destination and method as well as the communication time windows and the handling of retries.

In version 1 the possibility of data protection has been added offering the same options as defined in the “Data protection” IC.

This version of the interface class is intended to facilitate the use of relative and absolute data selection. It would be common practice to use an instance of the Push setup IC with relative data selection for routine data push, and an instance of the Push setup IC with absolute data for special cases.

The push takes place upon invoking the push method, triggered by a Push “Single action schedule” object, by an Alarm “Register monitor” object, by a dedicated internal event or externally. After the push operation has been triggered, it is executed according to the settings made in the given “Push setup” object. Depending on the communication window settings, the push is executed immediately or as soon as a communication window becomes active, after a random delay. If the push was not successful, retries are made. Push windows, delays and retries are shown in Figure 30.

Push setup		**0...n**	**class_id = 40, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. push_object_list	(static)	array				x + 0x08
3. send_destination_and_method	(static)	structure				x + 0x10
4. communication_window	(static)	array				x + 0x18
5. randomisation_start_interval	(static)	long-unsigned				x + 0x20
6. number_of_retries	(static)	unsigned				x + 0x28
7. repetition_delay	(static)	long-unsigned				x + 0x30
8. port_reference	(static)	octet-string				x + 0x38
9. push_client_SAP	(static)	integer				x + 0x40
10. push_protection_parameters	(static)	array				x + 0x48
Specific methods		***m/o***				
1. push (data)		m				x + 0x58

5.4.10.2 Attribute description

5.4.10.2.1 logical_name

Identifies the “Push setup” object instance. See 6.2.24.

5.4.10.2.2 push_object_list

Defines the list of attributes to be pushed.

Upon invocation of the *push* (data) method the items are sent to the destination defined in the *send_destination_and_method* attribute.

Two, mutually exclusive selective access mechanisms are available:

– relative selective access, i.e. entries defined relative to current date or entry are returned: this mechanism is controlled by the *data_index* element; or

– absolute selective access, i.e. entries in an explicitly defined date range or entry range are returned: entries selected by this mechanism are controlled by the *restriction_element* and the columns are controlled by the lower nibble of the MS byte in the *data_index*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
array push_object_definition

push_object_definition::=      structure

{
        class_id:          long-unsigned,
        logical_name:      octet-string,
        attribute_index:   integer,
        data_index:        long-unsigned,
        restriction:       restriction_element
}
```

Where:

– attribute_index is a pointer to the attribute within the object, identified by class_id and logical_name: attribute_index 1 refers to the first attribute (i.e. the logical_name), attribute_index 2 to the second attribute etc.; attribute_index 0 refers to all public attributes, noting IEC 62056-5-3:2023, 4.2.4.3.7;

– data_index is a pointer selecting a specific element of an attribute with a complex data type (structure or array).

– if the data type of the attribute is simple, then data_index has no meaning;

– if the data type of the attribute is a structure or an array – other than the buffer of a Profile generic object – then data_index points to one or several specific elements in the structure or array;

– when the attribute is the *buffer* of a “Profile generic” object, the data_index carries selective access parameters relative to current date or entry.

data_index	MS-Byte		LS-Byte
	Upper nibble	Lower nibble	

– 0x0000 = the whole attribute is pushed.

– 0x0001 to 0x0FFF = identifies one element in the complex attribute.

– 0x1000 to 0xFFFF = selective access to the array holding the buffer of a “Profile generic” object. The data-index selects entries within a number of last (recent) time periods, or a number of last (recent) entries, as well as the columns in the array.

The encoding is specified in Table 48.

When the attribute is the buffer of a “Profile generic” object, then restriction_element specifies absolute selective access parameters in an explicitly defined date range or entry range.

```
restriction_element::= structure

{
restriction_type:  enum:
                        (0) none,
                        (1) range by date,
                        (2) range by entry

restriction_value: CHOICE

        {
                null-data,                  // no restrictions apply
                restriction_by_date,
                restriction_by_entry
        }
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
restriction_by_date::= structure
{
        from_date: octet-string,
        to_date: octet-string
}

restriction_by_entry::= structure

{
        from_entry: double-long-unsigned,
        to_entry: double-long-unsigned
}
```

– restriction_element defines absolute selective access to a “Profile generic” buffer by date range (from_date to to_date) or by entries (from_entry to to_entry). To use this absolute selective access mechanism, data_index shall be:

– MS Byte upper nibble set to 0x0;

– MS Byte lower nibble set to 0x0 to 0xF in accordance with Table 48;

– lower byte set to 0x00;

– restriction_element is composed of restriction_type and restriction_value:

  - for restriction by date range the restriction_type element holds (1) restriction by date and the restriction_value element holds restriction_by_date structure;
  - for restriction by entries the restriction_type element holds (2) restriction by entry and the restriction_value element holds restriction_by_entry structure;
  - otherwise the restriction_type element holds (0) none and the restriction_value element holds null-data. This choice shall be taken also if relative selective access is to be used.

NOTE 1 If the push_object_list array is empty, the push operation is disabled.

NOTE 2 The push_object_list attribute itself can be also pushed to identify the data pushed.

Similarly to the case of the “Profile generic” IC, all attributes included in the push_object_list attribute are pushed regardless of the access rights to them. Therefore, writing of the push_object_list attribute should be restricted to clients with appropriate access rights.

5.4.10.2.3 send_destination_and_method

Contains the destination address (e.g. phone number, email address, IP address) where the data specified by the push_object_list has to be sent, as well as the sending method.

```
send_destination_and_method::= structure

{
        transport_service:      transport_service_type,
        destination:            octet-string,
        message:                message_type
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Where:

– the transport_service element defines the type of service used to push the data:

transport_service_type::= enum:

(0) TCP,
(1) UDP,
(2) reserved for FTP,
(3) reserved for SMTP,
(4) SMS,
(5) HDLC,
(6) reserved for M-Bus,
(7) reserved for ZigBee®,
(8) DLMS Gateway
(200...255) manufacturer specific

– the destination element contains the target address where the data has to be sent. The elements of the target address depend on the transport service used.

Each “Push setup” object instance specifies a single destination. If it is required to push data to several destinations, several “Push setup” objects have to be instantiated,

– the message_type element identifies the encoding of the xDLMS APDU used.

message_type::= enum:

(0) A-XDR encoded xDLMS APDU,
(1) XML encoded xDLMS APDU,
(128...255) manufacturer specific

– all other transport_service_type and message_type values are reserved for future use.

5.4.10.2.4 communication_window

Defines the time points when the communication window(s) for the push become(s) active (start_time) and inactive (end_time). See Figure 30.

```
array        window_element

window_element::= structure

{
      start_time:  octet-string,
      end_time:    octet-string
}
```

start_time and end_time are formatted as specified in 4.1.6.1 for *date-time* including wildcards.

If the end of a communication window is reached an already started push operation is completed.

If no communication windows are defined (array [0]) the push operation is always possible.

5.4.10.2.5 randomisation_start_interval

To avoid simultaneous push operations by a lot of devices at exactly the same point in time, a randomisation interval – in seconds – can be defined. This means that the push operation is not started immediately after the invocation of the *push* method but randomly delayed within the interval.

The *randomisation_start_interval* attribute defines the maximum value which can be obtained from a randomizing algorithm. The resulting value is used to delay the first push operation. It is not used anymore in case of retries.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

If no communication window is active when invoking the *push* method, the delay starts at the beginning of the next communication window. If the *push* method is invoked during an active communication window, the push operation is still delayed.

The *randomisation_start_interval* is only active for the first push attempt. If no *communication_window* is defined, the *randomisation_start_interval* is active for every push attempt.

If the *randomisation_start_interval* is set to 0 no delay is active.

If the randomisation time is longer than the first communication window, the first push attempt will be made during any later window.

5.4.10.2.6 number_of_retries

See 5.4.9.2.6.

5.4.10.2.7 repetition_delay

See 5.4.9.2.7.

5.4.10.2.8 port_reference

Contains the logical name of a communication port setup object allowing the selection of a specific communication channel for the push based on the transport_service_type. This mainly applies in cases where several channels of the same type are supported.

If this information is not available or not needed, the attribute may be left empty (octet-string [0]).

5.4.10.2.9 push_client_SAP

Defines the client SAP where the push is directed to in the supporting layer of the DataNotification service. The push process takes place within the application context of the AA which is linked to the *push_client_SAP* attribute. The security context is determined by the "Security setup" object referenced in the related "Association" object.

5.4.10.2.10 push_protection_parameters

Specifies all protection parameters to be applied to the DataNotification APDU when the push data is sent. It offers the same options as the "Data Protection" IC but it is bound to the sending of the data defined in the *push_object_list* attribute.

```
array protection_parameters_element

protection_parameters_element::= structure

{
protection_type: enum:
        (0) authentication,
        (1) encryption,
        (2) authentication and encryption,
        (3) digital signature

protection_options: structure

{
    transaction_id:          octet-string,
    originator_system_title: octet-string,
    recipient_system_title:  octet-string,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
                other_information:      octet-string,
                key_info:               key_info_element
            }
        }
```

Where:

– transaction_id holds the identifier of the transaction;

– originator_system_title holds the system title of the originator that applies the protection;

– recipient_system_title holds the system title of the recipient which will check and remove the given protection;

– other_information carries other information. Its contents may be specified in project specific companion specifications. An octet-string of length 0 indicates that this field is not used; key_info holds the information necessary for the recipient to obtain the right key for checking and removing authentication and encryption. In the case of digital signature, key_info is not necessary and it shall be a structure of 0 elements.

The fields transaction-id ….other-information are A-XDR encoded OCTET STRINGs. The length and the value of each field are included in the AAD when applicable.

```
key_info_element::= structure
{
    key_info_type: enum:

                (0) identified_key,
                -- used with identified_key_info_options
                (1) wrapped_key,
                -- used with wrapped_key_info_options
                (2) agreed_key
                -- used with agreed_key_info_options

    key_info_options: CHOICE

    {
        identified_key_info_options,
        wrapped_key_info_options,
        agreed_key_info_options
    }
}

identified_key_info_options::= enum:

                                (0) global_unicast_encryption_key,
                                (1) global_broadcast_encryption_key

wrapped_key_info_options::= structure
{
    kek_id:     enum:

                        (0) master_key,
    key_ciphered_data: octet-string
}

agreed_key_info_options::= structure
{
    key_parameters: octet-string,
    key_ciphered_data: octet-string
}
```

This attribute is first written by the client. The server may need to fill in some elements.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

For the use of the various elements see Table 22 and Table 23 of the ”Data protection” IC (class_id = 30, version = 0).

5.4.10.3 Method description

5.4.10.3.1 push (data)

See 5.4.9.3.1.

5.5 Previous versions of interface classes for time- and event-bound control

5.5.1 Parameter monitor (class_id = 65, version = 0)

5.5.1.1 Overview

Instances of the “Parameter monitor” IC monitor a list of COSEM object attributes holding parameters.

The parameters can be changed as usual. If the value of an attribute changes and this attribute is present in the *parameter_list* attribute, the identifier and the value of that attribute is automatically captured to the *changed_parameter* attribute. The time when the change of the parameter occurred is captured in the *capture_time* attribute. These attributes may be captured then by a “Profile generic” object. In this way, a log of all parameter changes can be built. For the OBIS code of the Parameter monitor log objects, see IEC 62056-6-1:2023, 6.5.

NOTE 1 In the case of simultaneous or quasi simultaneous parameter changes the order of capturing and logging the changed parameters has to be managed by the application.

Several “Parameter monitor” objects and corresponding “Profile generic” objects can be instantiated to manage a number of parameter groups. The link between the “Parameter monitor” object and the corresponding “Profile generic” object is via the *capture_object* attribute of the “Profile generic” object.

NOTE 2 As the various parameters may be of different type and length, the entries in the profile column holding the parameters will be also of different type and length. This can be managed for example by capturing different kind of parameters into different Parameter list “Profile generic” objects and parameter logs.

NOTE 3 The “Profile generic” object holding the parameter change log may capture other suitable object attributes, like the *time* attribute of the “Clock” object, and any other relevant values.

Parameter monitor	**0...n**	**class_id = 65, version = 0**			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. changed_parameter	structure				x + 0x08
3. capture_time	date-time				x + 0x10
4. parameter_list	array				x + 0x18
Specific methods	***m/o***				
1. add_parameter (data)	o				x + 0x20
2. delete_parameter (data)	o				x + 0x28

5.5.1.2 Attribute description

5.5.1.2.1 logical_name

Identifies the “Parameter monitor” object instance. See 6.2.14.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.5.1.2.2 changed_parameter

Holds the identifier and the value of the most recently changed parameter.

```
structure
{
    class_id:          long-unsigned,
    logical_name:      octet-string,
    attribute_index:   integer,
    attribute_value:   CHOICE
    -- CHOICE as specified in the “Data” interface class
}
```

5.5.1.2.3 capture_time

Provides data and time information showing when the value of the *changed_parameter* attribute has been captured.

date-time is formatted as specified in 4.1.6.1.

5.5.1.2.4 parameter_list

Holds the list of parameters managed by a given instance of the “Parameter monitor” IC.

```
parameter_list::= array parameter_list_element

parameter_list_element::= structure
{
    class_id:          long-unsigned,
    logical_name:      octet-string,
    attribute_index:   integer
}
```

NOTE The list of parameters monitored may be changed by using the *add_parameter* or *delete_parameter* methods or writing this attribute.

5.5.1.3 Method description

5.5.1.3.1 add_parameter (data)

Adds one parameter to the *parameter_list*.

```
data::= parameter_list_element
```

NOTE A parameter can be logged as soon as it is added to the list. Adding an element to the list does not affect the *buffer* of the “Profile generic” object capturing the *changed_parameter* attribute.

5.5.1.3.2 delete_parameter (data)

Deletes one parameter from the *parameter_list*.

```
data::= parameter_list_element
```

NOTE When a parameter is deleted from the parameter list, its changes will not be logged any more. Removing an element from the list does not affect the *buffer* of the “Profile generic” object capturing the *changed_parameter* attribute.

5.6 Previous versions of payment metering related interface classes

There are no previous versions to report.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7 Previous versions of interface classes for setting up data exchange via local ports and modems

5.7.1 IEC local port setup (class_id = 19, version = 0)

5.7.1.1 Overview

Instances of this IC define the operational parameters for communication using IEC 62056-21:2002. Several ports can be configured.

IEC local port setup		0...n	class_id = 19, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. default_mode	(static)	enum				x + 0x08
3. default_baud	(static)	enum				x + 0x10
4. prop_baud	(static)	enum				x + 0x18
5. response_time	(static)	enum				x + 0x20
6. device_addr	(static)	octet-string				x + 0x28
7. pass_p1	(static)	octet-string				x + 0x30
8. pass_p2	(static)	octet-string				x + 0x38
9. pass_w5	(static)	octet-string				x + 0x40
Specific methods		***m/o***				

5.7.1.2 Attribute description

5.7.1.2.1 logical_name

Identifies the “IEC local port setup” object instance. See 6.2.18.

5.7.1.2.2 default_mode

Defines the protocol used by the meter on the port.

enum:

(0) protocol according to IEC 62056-21:2002 (modes A…E)
(1) protocol according to IEC 62056-46:2002/AMD1:2006, Clause 8. Using this enumeration value all other attributes of this IC are not applicable.

5.7.1.2.3 default_baud

Defines the baud rate for the opening sequence

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(9) 115 200 baud

5.7.1.2.4 prop_baud

Defines the baud rate to be proposed by the meter.

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,
(9) 115 200 baud

5.7.1.2.5 response_time

Defines the minimum time between the reception of a request (end of request telegram) and the transmission of the response (begin of response telegram).

enum:

(0) 20 ms,
(1) 200 ms

5.7.1.2.6 device_addr

Device address according to IEC 62056-21:2002.

5.7.1.2.7 pass_p1

Password 1 according to IEC 62056-21:2002.

5.7.1.2.8 pass_p2

Password 2 according to IEC 62056-21:2002.

5.7.1.2.9 pass_w5

Password W5 reserved for national applications.

5.7.2 IEC HDLC setup, (class_id = 23, version = 0)

5.7.2.1 Overview

An instance of the “IEC HDLC setup” contains all data necessary to set up a communication channel according to IEC 62056-46:2002/AMD1:2006. Several communication channels can be configured.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC HDLC setup		0...n	class_id = 23, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	9	5	x + 0x08
3. window_size_transmit	(static)	unsigned	1	7	1	x + 0x10
4. window_size_receive	(static)	unsigned	1	7	1	x + 0x18
5. max_info_field_length_ transmit	(static)	unsigned	32	128	128	x + 0x20
6. max_info_field_length_ receive	(static)	unsigned	32	128	128	x + 0x28
7. inter_octet_time_out	(static)	long-unsigned	20	1000	25	x + 0x30
8. inactivity_time_out	(static)	long-unsigned	0		120	x + 0x38
9. device_address	(static)	long-unsigned	0x0010	0x3FFD		x + 0x40
Specific methods		***m/o***				

5.7.2.2 Attribute description

5.7.2.2.1 logical_name

Identifies the “IEC HDLC setup” object instance. See 6.2.20.

5.7.2.2.2 comm_speed

The communication speed supported by the corresponding port:

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,
(9) 115 200 baud

This communication speed can be overridden if the HDLC mode of a device is entered through a special mode of another protocol.

5.7.2.2.3 window_size_ transmit

The maximum number of frames that a device or system can transmit before it needs to receive an acknowledgement from a corresponding station. During logon, other values can be negotiated.

5.7.2.2.4 window_size_ receive

The maximum number of frames that a device or system can receive before it needs to transmit an acknowledgement to the corresponding station. During logon, other values can be negotiated.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.2.2.5 max_info_length_ transmit

The maximum information field length that a device can transmit. During logon, a smaller value can be negotiated.

5.7.2.2.6 max_info_length_ receive

The maximum information field length that a device can receive. During logon, a smaller value can be negotiated.

5.7.2.2.7 inter_octet_time_out

Defines the time, expressed in milliseconds, over which, when no character is received from the primary station, the device will treat the already received data as a complete frame.

5.7.2.2.8 inactivity_time_out

Defines the time, expressed in seconds over which, when no frame is received from the primary station, the device will process a disconnection.

When this value is set to 0, this means that the *inactivity_time_out* is not operational.

5.7.2.2.9 device_address

Contains the physical device address of a device:

In the case of single byte addressing:

0x00	NO_STATION Address,
0x01…0x0F	Reserved for future use,
0x10…0x7D	Usable address space,
0x7E	'CALLING' device address,
0x7F	Broadcast address

In the case of double byte addressing:

0x0000	NO_STATION address,
0x0001…0x000F	Reserved for future use,
0x0010…0x3FFD	Usable address space,
0x3FFE	CALLING' physical device address,
0x3FFF	Broadcast address

5.7.3 IEC twisted pair (1) setup (class_id = 24, version = 0)

5.7.3.1 Overview

NOTE The use of version 0 of the IEC twisted pair (1) setup IC is deprecated.

This IC allows modelling and configuring communication channels according to IEC 62056-31:1999. Several communication channels can be configured.

MAC address

Secondary stations – typically metering end devices – hold a secondary station address (ADS). The length of the ADS shall be 6 octets and consists of the elements shown in Table 49. This address shall be worldwide unique and it shall be assigned by the manufacturer to the secondary station. The ADS assigned is valid through the lifetime of the secondary station.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 49 – ADS address elements

Elements of ADS	Description	Length (byte)	Type	Range
manufacturer_id	Assigned upon request by the Euridis Association to the manufacturer. It shall be used in all devices using the *Twisted pair with carrier signalling* medium and made by this manufacturer.	1	BCD	0-99
year_of_manufacture	Holds the year of manufacturing the device (the lower two numbers only).	1		0-99
equipment_id	Related to the type of equipment concerned and provides information on the functionality available. Like the manufacturer_id, it is assigned by the Euridis Association.	1		0-99
device_serial_number	The manufacturing number of the device assigned by the manufacturer. It should have the same value as the value held by the object Device ID 1, see IEC 62056-6-1:2023, clause 6.	3		000001-999999 000000 is reserved
EXAMPLE 031267123456: 03: ITRON International; 12: Year 2012; 67: Smart meter LINKY Single phase 123456: Serial number beginning each year from 000001.				

IEC twisted pair (1) setup		0...n	class_id = 24, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. secondary_address	(static)	octet-string				x + 0x08
3. primary_address_list	(static)	primary_address_list_type				x + 0x10
4. tabi_list	(static)	tabi_list_type				x + 0x18
5. fatal_error	(dyn.)	enum				x + 0x20
Specific methods		***m/o***				

5.7.3.2 Attribute description

5.7.3.2.1 logical_name

Identifies the “IEC twisted pair setup” object instance. See 6.2.21.

5.7.3.2.2 secondary_address

Secondary_address memorizes the ADS of the secondary station that corresponds to the real equipment.

octet-string (SIZE(6))

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.3.2.3 primary_address_list

Primary_address_list memorizes the list of ADP or primary station physical addresses for which each logical device of the real equipment (the secondary station) has been programmed.

primary_address_list_type::= array primary_address_element

primary_address_element::= octet-string

The length of the octet-string is one octet.

5.7.3.2.4 tabi_list

tabi_list represents the list of the TAB(i) for which the real equipment (the secondary station) has been programmed in case of forgotten station call.

tabi_list_type::= array tabi_element

tabi_element::= integer

5.7.3.2.5 fatal_error

fatal_error represents the last occurrence of one of the fatal errors of the protocols described in IEC 62056-31:1999.

The initial default value of this variable is 0x00. Then, each fatal error is spotted.

enum:

(0) No-error,
(1) t-EP-1F,
(2) t-EP-2F,
(3) t-EL-4F,
(4) t-EL-5F,
(5) eT-1F,
(6) eT-2F,
(7) e-EP-3F,
(8) e-EP-4F,
(9) e-EP-5F,
(10) e-EL-2F

5.7.4 PSTN modem configuration (class_id = 27, version = 0)

5.7.4.1 Overview

NOTE The name of this IC was changed to “Modem configuration” in Edition 2.

An instance of the “PSTN modem configuration” IC stores data related to the initialization of modems, which are used for data transfer from/to a device. Several modems can be configured.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PSTN modem configuration		0...n	class_id = 27, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	9	5	x + 0x08
3. initialization_string	(static)	array				x + 0x10
4. modem_profile	(static)	array				x + 0x18
Specific methods		***m/o***				

5.7.4.2 Attribute description

5.7.4.2.1 logical_name

Identifies the “PSTN modem configuration” object instance. See 6.2.6.

5.7.4.2.2 comm_speed

The communication speed between the device and the modem, not necessarily the communication speed on the WAN.

enum:

(0) 300 baud,
(1) 600 baud,
(2) 1 200 baud,
(3) 2 400 baud,
(4) 4 800 baud,
(5) 9 600 baud,
(6) 19 200 baud,
(7) 38 400 baud,
(8) 57 600 baud,
(9) 115 200 baud

5.7.4.2.3 initialization_string

This data contains all the necessary initialization commands to be sent to the modem in order to configure it properly. This may include the configuration of special modem features.

```
array   initialization_string_element

initialization_string_element::= structure
{
        request:    octet-string,
        response:   octet-string
}
```

If the array contains more than one initialization string element, they are subsequently sent to the modem after receiving an answer matching the defined response.

REMARK It is assumed that the modem is pre-configured so that it accepts the initialization_string. If no initialization is needed, the initialization string is empty.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.4.2.4 modem_profile

This data defines the mapping from Hayes standard commands/responses to modem specific strings.

array modem_profile_element

modem_profile_element: octet-string

The *modem_profile* array shall contain the corresponding strings for the modem used in following order:

Element 0: OK,
Element 1: CONNECT,
Element 2: RING,
Element 3 NO CARRIER,
Element 4: ERROR,
Element 5: CONNECT 1 200,
Element 6 NO DIAL TONE,
Element 7: BUSY,
Element 8: NO ANSWER,
Element 9: CONNECT 600,
Element 10: CONNECT 2 400,
Element 11: CONNECT 4 800,
Element 12 CONNECT 9 600,
Element 13: CONNECT 14 400,
Element 14: CONNECT 28 800,
Element 15: CONNECT 36 600,
Element 16: CONNECT 56 000

5.7.5 Auto answer (class_id = 28, version = 0)

5.7.5.1 Overview

This IC allows modelling how the device manages the “Auto answer” function of the modem, i.e. answering of incoming calls. Several modems can be configured.

Auto answer		**0...n**	**class_id = 28, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. listening_window	(static)	array				x + 0x10
4. status	(dyn.)	enum				x + 0x18
5. number_of_calls	(static)	unsigned				x + 0x20
6. number_of_rings	(static)	nr_rings_type				x + 0x28
Specific methods		***m/o***				

5.7.5.2 Attribute description

5.7.5.2.1 logical_name

Identifies the “Auto answer” object instance. See 6.2.6.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.5.2.2 mode

Defines the working mode of the line when the device is auto answer.

enum:

(0) line dedicated to the device,

(1) shared line management with a limited number of calls allowed. Once the number of calls is reached, the window status becomes inactive until the next start date, whatever the result of the call,

(2) shared line management with a limited number of successful calls allowed. Once the number of successful communications is reached, the window status becomes inactive until the next start date,

(3) currently no modem connected,

(200...255) manufacturer specific modes

5.7.5.2.3 listening_window

Contains the start and end instant when the window becomes active (for the start instant), and inactive (for the end instant). The start_date defines implicitly the period.

EXAMPLE When the day of month is not specified (equal to 0xFF) this means that we have a daily share line management. Daily, monthly ...window management can be defined.

```
array   window_element

window_element::= structure
{
        start _time: octet-string,
        end_time:   octet-string
}
```

start_time and end_time are formatted as specified in 4.6.1 for *date-time*.

5.7.5.2.4 status

Here is defined the status of the window.

enum:

(0) Inactive: the device will manage no new incoming call. This status is automatically reset to Active when the next listening window starts,

(1) Active: the device can answer to the next incoming call,

(2) Locked: This value can be set automatically by the device or by a specific client when this client has completed its reading session and wants to give the line back to the customer before the end of the window duration. This status is automatically reset to Active when the next listening window starts.

5.7.5.2.5 number_of_calls

This number is the reference used in modes 1 and 2.

When set to 0, this means there is no limit.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.5.2.6 number_of_rings

Defines the number of rings before the meter connects the modem. Two cases are distinguished: number of rings within the window defined by attribute *listening_window* and number of rings outside the *listening_window*.

```
nr_rings_type::= structure
{
   nr_rings_in_window:          unsigned, (0: no connect in window)
        nr_rings_out_of_window: unsigned (0: no connect out of window)
}
```

5.7.6 PSTN auto dial (class_id = 29, version = 0)

5.7.6.1 Overview

NOTE The name of this IC was changed to “Auto connect” in Edition 2.

An instance of the “PSTN auto dial” IC stores data related to the management data transfer between the device and the modem to perform auto dialling. Several modems can be configured.

PSTN auto dial		**0...n**	**class_id = 29, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. repetitions	(static)	unsigned				x + 0x10
4. repetition_delay	(static)	long-unsigned				x + 0x18
5. calling_window	(static)	array				x + 0x20
6. phone_list	(static)	array				x + 0x28
Specific methods		***m/o***				

5.7.6.2 Attribute description

5.7.6.2.1 logical_name

Identifies the “PSTN auto dial” object instance. See 6.2.6.

5.7.6.2.2 mode

Defines if the device can perform auto-dialling.

enum:

(0) no auto dialling,
(1) auto dialling allowed anytime,
(2) auto dialling allowed within the validity time of the calling window,
(3) “regular” auto dialling allowed within the validity time of the calling window; “alarm” initiated auto dialling allowed anytime,
(200…255) manufacturer specific modes

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.6.2.3 repetitions

The maximum number of trials In the case of unsuccessful dialling attempts.

5.7.6.2.4 repetition_delay

The time delay, expressed in seconds until an unsuccessful dial attempt can be repeated.

repetition_delay 0 means delay is not specified

5.7.6.2.5 calling_window

Contains the start and end date/time stamp when the window becomes active (for the start instant), or inactive (for the end instant). The start_date defines implicitly the period.

EXAMPLE When day of month is not specified (equal to 0xFF) this means that we have a daily share line management. Daily, monthly …window management can be defined.

```
array   window_element
window_element::= structure
{
        start_time:  octet-string,
        end_time:    octet-string
}
```

start_time and end_time are formatted as specified in 4.6.1 for *date-time*.

5.7.6.2.6 phone_list

Contains phone numbers, the device modem has to call under certain conditions. The link between entries in the array and the conditions are not contained in here.

```
array   phone_number
phone_number::= octet-string
```

5.7.7 Auto connect (class_id = 29, version = 1)

5.7.7.1 Overview

This IC allows modelling the management of data transfer from the device to one or several destinations.

The messages to be sent, the conditions on which they shall be sent and the relation between the various modes, the calling windows and destinations are not defined here.

Depending on the mode, one or more instances of this IC may be necessary to perform the function of sending out messages.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Auto connect		0...n	class_id = 29, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. repetitions	(static)	unsigned				x + 0x10
4. repetition_delay	(static)	long-unsigned				x + 0x18
5. calling_window	(static)	array				x + 0x20
6. destination_list	(static)	array				x + 0x28
Specific methods		***m/o***				

5.7.7.2 Attribute description

5.7.7.2.1 logical_name

See 5.7.6.2.1.

5.7.7.2.2 mode

Defines the mode controlling the auto dial functionality concerning the timing, the message type to be sent and the infrastructure to be used.

enum:

(0) no auto dialling,
(1) auto dialling allowed anytime,
(2) auto dialling allowed within the validity time of the calling window,
(3) “regular” auto dialling allowed within the validity time of the calling window; “alarm” initiated auto dialling allowed anytime,
(4) SMS sending via Public Land Mobile Network (PLMN),
(5) SMS sending via PSTN,
(6) email sending,
(200..255) manufacturer specific modes

5.7.7.2.3 repetitions

See 5.7.6.2.3.

5.7.7.2.4 repetition_delay

See 5.7.6.2.4.

5.7.7.2.5 calling_window

See 5.7.6.2.5.

5.7.7.2.6 destination_list

Contains the list of destinations (for example phone numbers, email addresses or their combinations) where the message(s) have to be sent under certain conditions. The conditions and their link to the elements of the array are not defined here.

array destination

destination::= octet-string

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.8 GSM diagnostic (class_id = 47, version = 0)

5.7.8.1 Overview

The GSM/GPRS network is undergoing constant changes in terms of registration status, signal quality etc. It is necessary to monitor and log the relevant parameters in order to obtain diagnostic information that allows identifying communication problems in the network.

An instance of the “GSM diagnostic” class stores parameters of the GSM/GPRS network necessary for analysing the operation of the network.

A GSM diagnostic “Profile generic” object is also available to capture the attributes of the GSM diagnostic object, see IEC 62056-6-1:2023, 6.5.

GSM diagnostic		**0…n**	**class_id = 47, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. operator	(dyn.)	visible-string				x + 0x08
3. status	(dyn.)	enum	0	255	0	x + 0x10
4. cs_attachment	(dyn.)	enum	0	255	0	x + 0x18
5. ps_status	(dyn)	enum	0	255	0	x + 0x20
6. cell_info	(dyn.)	cell_info_type				x + 0x30
7. adjacent_cells	(dyn.)	array				x + 0x38
8. capture_time	(dyn.)	date-time				x + 0x40
Specific methods		***m/o***				

5.7.8.2 Attribute description

5.7.8.2.1 logical_name

Identifies the “GSM Diagnostic” object instance. For logical names, see 6.2.23.

5.7.8.2.2 operator

Holds the name of the network operator e.g. “YourNetOp”

5.7.8.2.3 status

Indicates the registration status of the modem.

enum:

(0) not registered,
(1) registered, home network,
(2) not registered, but MT is currently searching a new operator to register to,
(3) registration denied,
(4) unknown,
(5) registered, roaming
(6) ... (255) reserved

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.8.2.4 cs_attachment

Indicates the current circuit switched status.

enum:

(0) inactive,
(1) incoming call,
(2) active,
(3) ... (255) reserved

5.7.8.2.5 ps_status

The ps_status value field indicates the packet switched status of the modem.

enum:

(0) inactive,
(1) GPRS,
(2) EDGE,
(3) UMTS,
(4) HSDPA,
(5) ... (255) reserved

5.7.8.2.6 cell_info

Represents the cell information:

```
cell_info_type::= structure
    {
        cell_ID:          long-unsigned,
        location_ID:      long-unsigned,
        signal_quality:   unsigned,
        ber:              unsigned
    }
```

Where:

– cell_ID: Two-byte cell ID in hexadecimal format;
– location_ID: Two-byte location area code (LAC) in hexadecimal format (e.g. "00C3" equals 195 in decimal);
– signal_quality: Represents the signal quality:

(0) –113 dBm or less,
(1) –111 dBm,
(2...30) –109...-53 dBm,
(31) –51 or greater,
(99) not known or not detectable;

– ber: Bit Error Rate (BER) measurement in percent:

(0...7) as RXQUAL_n values specified in ETSI GSM 05.08:1996,8.2.4.
(99) not known or not detectable.

5.7.8.2.7 adjacent_cells

```
array          adjacent_cell_info

adjacent_cell_info::= structure
{
   cell_ID:          long-unsigned,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

signal_strength: unsigned

}

Where:

– cell_ID: Two-byte cell ID in hexadecimal format;

– signal_quality: Represents the signal quality:

(0) –113 dBm or less,
(1) –111 dBm,
(2...30) –109...-53 dBm,
(31) –51 or greater,
(99) not known or not detectable.

5.7.8.2.8 capture_time

Holds the date and time when the data have been last captured.

date-time is formatted as specified in 4.1.6.1.

5.7.9 GSM diagnostic (class_id: 47, version: 1)

5.7.9.1 Overview

The cellular network is undergoing constant changes in terms of registration status, signal quality, etc. It is necessary to monitor and log the relevant parameters in order to obtain diagnostic information that allows identifying communication problems in the network.

An instance of the "GSM diagnostic" class stores parameters of the GSM/GPRS, UMTS, CDMA or LTE network necessary for analysing the operation of the network.

A GSM diagnostic "Profile generic" object is also available to capture the attributes of the GSM diagnostic object, see IEC 62056-6-1:2023, 6.5.

GSM diagnostic		**0...n**	**class_id = 47, version = 1**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. operator	(dyn.)	visible-string				x + 0x08
3. status	(dyn.)	enum	0	255	0	x + 0x10
4. cs_attachment	(dyn.)	enum	0	255	0	x + 0x18
5. ps_status	(dyn)	enum	0	255	0	x + 0x20
6. cell_info	(dyn.)	cell_info_type				x + 0x30
7. adjacent_cells	(dyn.)	array				x + 0x38
8. capture_time	(dyn.)	date-time				x + 0x40
Specific methods		***m/o***				

5.7.9.2 Attribute description

5.7.9.2.1 logical_name

Identifies the "GSM Diagnostic" object instance. For logical names, see 6.2.23.

5.7.9.2.2 operator

Holds the name of the network operator e.g. "YourNetOp".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.9.2.3 status

Indicates the registration status of the modem.

enum:

(0) not registered,
(1) registered, home network,
(2) not registered, but MT is currently searching a new operator to register to,
(3) registration denied,
(4) unknown,
(5) registered, roaming
(6) ... (255) reserved

5.7.9.2.4 cs_attachment

Indicates the current circuit switched status.

enum:

(0) inactive,
(1) incoming call,
(2) active,
(3) ... (255) reserved.

5.7.9.2.5 ps_status

The *ps_status* value field indicates the packet switched status of the modem.

enum:

(0) inactive,
(1) GPRS,
(2) EDGE,
(3) UMTS,
(4) HSDPA,
(5) LTE,
(6) CDMA,
(7)...(255) reserved

5.7.9.2.6 cell_info

Represents the cell information:

```
cell_info_type::= structure
{
    cell_ID:            double-long-unsigned,
    location_ID:        long-unsigned,
    signal_quality:     unsigned,
    ber:                unsigned,
    mcc:                long-unsigned,
    mnc:                long-unsigned,
    channel_number:     double-long-unsigned
}
```

Where:

– cell_ID: Four-byte cell ID in hexadecimal format;

– location_ID: Two-byte location area code (LAC) in the case of GSM networks or Tracking Area Code (TAC) in the case of UMTS, CDMA or LTE networks in hexadecimal format (e.g. "00C3" equals 195 in decimal);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– signal_quality: Represents the signal quality:

(0) –113 dBm or less,
(1) –111 dBm,
(2...30) –109...-53 dBm,
(31) –51 or greater,
(99) not known or not detectable;

– ber: Bit Error Rate (BER) measurement in percent:

(0...7) as RXQUAL_n values specified in ETSI GSM 05.08:1996, 8.2.4
(99) not known or not detectable.

– mcc: Mobile Country Code of the serving network, as defined in ITU-T E.212 (05.2008);

– mnc: Mobile Network Code of the serving network, as defined in ITU-T E.212 (05.2008);

– channel_number: Represents the absolute radio-frequency channel number (ARFCN or eaRFCN for LTE network).

5.7.9.2.7 adjacent_cells

```
array        adjacent_cell_info

adjacent_cell_info::= structure
{
     cell_ID:            double-long-unsigned,
     signal_quality:          unsigned
}
```

Where:

– cell_ID: Four-byte cell ID in hexadecimal format;

– signal_quality: Represents the signal quality:

(0) –113 dBm or less,
(1) –111 dBm,
(2...30) –109...-53 dBm,
(31) –51 or greater,
(99) not known or not detectable.

5.7.9.2.8 capture_time

Holds the date and time when the data have been last captured.

date-time is formatted as specified in 4.1.6.1.

5.7.10 LTE monitoring (class_id: 151, version: 0)

5.7.10.1 Overview

Instances of the 'LTE monitoring" IC allow monitoring LTE modems by handling all data necessary data for this purpose.

LTE monitoring		**0...n**	**class_id = 151, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. lte_quality_of_service	(dyn.)	LTE_qos_type				x + 0x08
Specific methods		***m/o***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.7.10.2 Attribute description

5.7.10.2.1 loogical_name

Identifies the “LTE monitoring” object instance. See 6.2.23.

5.7.10.2.2 lte_quality_of_service

Represents the quality of service for the LTE network

```
LTE_qos_type::= structure
{
    T3402:          long-unsigned,
    T3412:          long-unsigned,
    RSRQ:           unsigned,
    RSRP:           unsigned,
    qRxlevMin:      integer
}
```

Where:

– T3402: timer in seconds, used on PLMN selection procedure and sent by the network to the modem. Refer to 3GPP TS 24.301 V13.4.0 (2016-01) for details;

– T3412: timer in seconds used to manage the periodic tracking area updating procedure and sent by the network to the modem. Refer to 3GPP TS 24.301 V13.4.0 (2016-01) for details;

– RSRQ: represents the signal quality as defined in 3GPP TS 24.301 V13.4.0 (2016-01):

(0) –19,5dB,
(1) –19 dB,
(2...31) –18,5…-3,5 dB,
(32) –3dB,
(99) Not known or not detectable;

– RSRP: represents the signal level as defined in 3GPP TS 24.301 V13.4.0 (2016-01):

(0) –140dBm,
(1) –139 dBm,
(2... 94) –138…-45 dBm,
(95) –44dBm,
(99) Not known or not detectable;

– qRxlevMin: specifies the minimum required Rx level in the cell in dBm as defined in 3GPP TS 24.301 V13.4.0 (2016-01).

5.8 Previous versions of interface classes for setting up data exchange via M-Bus

5.8.1 M-Bus client (class_id = 72, version = 0)

5.8.1.1 Overview

Instances of this IC allow setting up M-Bus slave devices and to exchange data with them. Each M-Bus client object controls one M-Bus slave device. For details on the M-Bus dedicated application layer, see EN 13757-3:2013.

The M-Bus client device may have one or more physical M-Bus interfaces, which can be configured using instances of the M-Bus master port setup IC, see 4.8.5.

An M-Bus slave device is identified with its Primary Address, Identification Number, Manufacturer ID etc. as defined in EN 13757-3:2013 Clause 5, *Variable Data respond*. These parameters are carried by the respective attributes of the M-Bus client IC, see 4.8.3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Values to be captured from an M-Bus slave device are identified by the *capture_definition* attribute, containing a list of data identifiers (DIB, VIB) for the M-Bus slave device. The data are captured periodically or on an appropriate trigger. Each data element is stored in an M-Bus value object, of IC "Extended register". M-Bus value objects may be captured in M-Bus Profile generic objects, eventually along with other, not M-Bus specific objects.

Using the methods of M-Bus client objects, M-Bus slave devices can be installed and de-installed. It is also possible to send data to M-Bus slave devices and to perform operations like resetting alarms, setting the clock, transferring an encryption key, etc.

M-Bus client		**0...n**	**class_id = 72, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name	(static)	octet-string				x
2. mbus_port_reference	(static)	octet-string				x + 0x08
3. capture_definition	(static)	array				x + 0x10
4. capture_period	(static)	double-long-unsigned				x + 0x18
5. primary_address	(dyn.)	unsigned				x + 0x20
6. identification_number	(dyn.)	double-long-unsigned				x + 0x28
7. manufacturer_id	(dyn.)	long-unsigned				x + 0x30
8. version	(dyn.)	unsigned				x + 0x38
9. device_type	(dyn.)	unsigned				x + 0x40
10. access_number	(dyn.)	unsigned				x + 0x48
11. status	(dyn.)	unsigned				x + 0x50
12. alarm	(dyn.)	unsigned				x + 0x58
Specific methods		***m/o***				
1. slave_install (data)		o				x + 0x60
2. slave_deinstall (data)		o				x + 0x68
3. capture (data)		o				x + 0x70
4. reset_alarm (data)		o				x + 0x78
5. synchronize_clock (data)		o				x + 0x80
6. data_send (data)		o				x + 0x88
7. set_encryption_key (data)		o				x + 0x90
8. transfer_key (data)		o				x + 0x98

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.8.1.2 Attribute description

5.8.1.2.1 logical_name

Identifies the "M-Bus client" object instance. See 6.2.22.

5.8.1.2.2 mbus_port_reference

Provides reference to an "M-Bus master port setup" object, used to configure an M-Bus port, each interface allowing to exchange data with one or more M-Bus slave devices.

5.8.1.2.3 capture_definition

Provides the capture_definition for M-Bus slave devices.

```
array        capture_definition_element
capture_definition_element::= structure
{
      data_information_block:          octet-string,
      value_information_block:         octet-string
}
```

NOTE The elements data_information_block and value_information_block correspond to Data Information Block (DIB) and Value Information Block (VIB) described in EN 13757-3:2013, 6.2 and Clause 7 respectively.

5.8.1.2.4 capture_period

>= 1: Automatic capturing assumed. Specifies the capture period in seconds.

0: No automatic capturing: capturing is triggered externally or capture events occur asynchronously.

5.8.1.2.5 primary_address

Carries the primary address of the M-Bus slave device, in the range 0...250.

If the slave device is already configured and thus, its primary address is different from 0, then this value shall be written to the *primary_address* attribute. From this moment, the data exchange with the M-Bus slave device is possible.

Otherwise, the *slave_install* method shall be used; see 5.8.1.3.1.

NOTE The *primary_address* attribute cannot be used to store a desired primary address for an unconfigured slave device. If the primary address attribute is set, this means that the M-Bus client can immediately operate with this primary address, which is not the case with an unconfigured slave device.

5.8.1.2.6 identification_number

Carries the Identification Number element of the data header as specified in EN 13757-3:2013, 5.4.

It is either a fixed fabrication number or a number changeable by the customer, coded with 8 BCD packed digits (4 Byte), and which thus runs from 00 000 000 to 99 999 999. It can be preset at fabrication time with a unique number, but could be changeable afterwards, especially if in addition a unique and not changeable fabrication number (DIF = 0x0C, VIF = 0x78 is provided.

5.8.1.2.7 manufacturer_id

Carries the Manufacturer Identification element of the data header as specified in EN 13757-3:2013, 5.5.

It is coded unsigned binary with 2 bytes. This *manufacturer_id* is calculated from the ASCII code of the IEC 62056-21 manufacturer ID (three uppercase letters), using the formula specified in EN 13757-3:2013, 5.5.

5.8.1.2.8 version

Carries the Version element of the data header as specified in EN 13757-3:2013, 5.6. It specifies the generation or version of the meter and depends on the manufacturer. It can be used to make sure, that within each version number the identification # is unique.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.8.1.2.9 device_type

Carries the Device type identification element of the data header as specified in EN 13757-3:2013, 5.7, Table 3.

5.8.1.2.10 access_number

Carries the Access Number element of the data header as specified in EN 13757-3:2013, 5.8.

It has unsigned binary coding, and it is incremented (modulo 256) by one before or after each RSP-UD from the slave. Since it can also be used to enable private end users to detect an unwanted over-frequently readout of their consumption meters, it should not be resettable by any bus communication.

5.8.1.2.11 status

Carries the Status byte element of the data header as specified in

EN 13757-3:2013, 5.9, Tables 4 and 5.

5.8.1.2.12 alarm

Carries the Alarm state specified in EN 13757-3:2013, Annex D. It is coded with data type D (Boolean, in this case 8 bit). Set bits signal alarm bits or alarm codes. The meaning of these bits is manufacturer specific.

5.8.1.3 Method description

5.8.1.3.1 slave_install (data)

Installs a slave device, which is yet unconfigured (its primary address is 0).

This method can be successfully invoked only if the value of the primary_address attribute is 0. The following actions are performed:

– the M-Bus address 0 is checked for presence of a new device.
– if no uninstalled M-Bus slave is found, the method invocation fails;
– if the *slave_install* method is invoked with no parameter, then the primary address is assigned automatically. This is done by checking the *primary_address* attribute of all M-Bus client objects in the DLMS®/COSEM device and then selecting the first unused number. The primary_address attribute is set to this address and it is then transferred to the M-Bus slave device;
– if the *slave_install* method is invoked with a primary address as a parameter, then the *primary_address* attribute is set to this value and it is then transferred to the M-Bus slave device.

data::= unsigned (no data, or a valid primary address)

NOTE Unconfigured slave devices are configured with primary address as specified in EN 13757-3:2013, Clause E.5.

5.8.1.3.2 slave_deinstall (data)

De-installs the slave device. The main purpose of this service is to uninstall the M Bus slave device and to prepare the master for the installation of a new device. The following actions are performed:

– the M-Bus address is set to 0 in the M-Bus slave device;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– the encryption key transferred previously to the M-Bus slave device is destroyed; the default key is not affected;
– the attribute *primary_address* is also set to 0.

NOTE A new M-Bus slave can be installed only, once the value of the primary_address attribute is 0.

data::= integer (0)

5.8.1.3.3 capture

Capture values – as specified by the *capture_definition* attribute – from the M-Bus slave device.

data::= integer (0)

5.8.1.3.4 reset_alarm

Reset alarm state of the M-Bus slave device.

data::= integer (0)

5.8.1.3.5 synchronize_clock

Synchronize the clock of the M-Bus slave device with that of the M-Bus client device.

data::= integer (0)

5.8.1.3.6 data_send

Send data to the M-Bus slave device.

```
data::=      array data_definition_element

data_definition_element::= structure
{
      data_information_block:          octet-string,
      value_information_block:         octet-string
      data:                            CHOICE
      {
            -- simple data types
            null-data                  [0],
            bit-string                 [4],
            double-long                [5],
            double-long-unsigned       [6],
            octet-string               [9],
            visible-string             [10],
            utf8-string                [12],
            integer                    [15],
            long                       [16],
            unsigned                   [17],
            long-unsigned              [18],
            long64                     [20],
            long64-unsigned            [21],
            float32                    [23],
            float64                    [24]
      }
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.8.1.3.7 set_encryption_key

Sets encryption key to be used with the M-Bus slave device.

Changing the encryption key requires two steps: First, the key is sent to the M-Bus slave, encrypted with the master key, using the *transfer_key* method. Second, the key is set in the M-Bus master using the *set_encryption_key* method.

data::= octet-string (encryption_key)

After the installation of the M-Bus slave, the M-Bus client holds an empty encryption key. With this, encryption of M-Bus telegrams is disabled. Encryption can be disabled by invoking the *set_encryption_key* method with null- data as a parameter.

5.8.1.3.8 transfer_key

Transfers an encryption key to the M-Bus slave.

data::= octet-string (encrypted_key)

Each M-Bus slave device shall be delivered with a default encryption key.

Before encrypted M-Bus telegrams can be used, an operational encryption key has to be sent to the M-Bus slave, by invoking the *transfer_key* method. The method invocation parameter is the operational encryption key encrypted with the default key of the M-Bus slave device. The M-Bus telegram sent is not encrypted. After the execution, the encryption is enabled and all further telegrams are encrypted.

A new encryption key can be set in the M-Bus client by invoking the *set_encryption_key* method with the new encryption key as a parameter.

With further invocations of the *transfer_key* method, new encryption keys can be sent to the M-Bus slave. The method invocation parameter is the new encryption key encrypted with the default key. The M-Bus telegram is encrypted.

When an M-Bus slave is de-installed, the encryption key is destroyed, but the default key is not affected. Encryption remains disabled until a new encryption is transferred.

5.9 Previous versions of interface classes for setting up data exchange over the internet

There are no previous versions to report.

5.10 Previous versions of interface classes for data exchange using S-FSK PLC

5.10.1 S-FSK Phy&MAC setup (class_id = 50, version = 0)

5.10.1.1 Overview

NOTE The use of this version is deprecated.

An instance of the “S-FSK Phy&MAC setup” IC stores the data necessary to set up and manage the physical and the MAC layer of the PLC S-FSK lower layer profile.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

S-FSK Phy&MAC setup		0...n	class_id = 50, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. initiator_electrical_phase	(static)	enum	0	3		x + 0x08
3. delta_electrical_phase	(dyn.)	enum	0	6		x + 0x10
4. max_receiving_gain	(static)	unsigned				x + 0x18
5. max_transmitting_gain	(static)	unsigned				x + 0x20
6. search_initiator_gain	(static)	unsigned				x + 0x28
7. frequencies	(static)	frequencies_type				x + 0x30
8. mac_address	(dyn.)	long-unsigned			FFE	x + 0x38
9. mac_group_addresses	(static)	array				x + 0x40
10. repeater	(static)	enum			1	x + 0x48
11. repeater_status	(dyn.)	boolean				x + 0x50
12. min_delta_credit	(dyn.)	unsigned				x + 0x58
13. initiator_mac_address	(dyn.)	long-unsigned				x + 0x60
14. synchronization_locked	(dyn.)	boolean				x + 0x68
Specific methods		***m/o***				

5.10.1.2 Attribute description

5.10.1.2.1 logical_name

Identifies the “S-FSK Phy&MAC setup” object instance. See 6.2.24.

5.10.1.2.2 initiator_electrical_phase

Holds the MIB variable initiator-electrical-phase (variable 18) specified in IEC 61334-4-512:2001, 5.8.

It is written by the client system to indicate the phase to which it is connected.

enum:

(0) Not defined (default),
(1) Phase 1,
(2) Phase 2,
(3) Phase 3.

NOTE This enumeration is different from that of IEC 61334-4-512:2001.

5.10.1.2.3 delta_electrical_phase

Holds the MIB variable delta-electrical-phase (variable 1) specified in IEC 61334-4-512:2001, 5.2 and IEC 61334-5-1:2001, 3.5.5.3.

It indicates the phase difference between the client's connecting phase and the server's connecting phase. The following values are predefined:

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

enum:

(0) Not defined: the server is temporarily not able to determine the phase difference,
(1) The server system is connected to the same phase as the client system.

The phase difference between the server's connecting phase and the client's connecting phase is equal to:

(2) 60 degrees,
(3) 120 degrees,
(4) 180 degrees,
(5) -120 degrees,
(6) -60 degrees.

5.10.1.2.4 max_receiving_gain

Holds the MIB variable max-receiving-gain (variable 2) specified in

IEC 61334-4-512:2001, 5.2 and in IEC 61334-5-1:2001, 3.5.5.3.

Corresponds to the maximum allowed gain bound to be used by the server system in the receiving mode. The default unit is dB.

NOTE In IEC 61334-4-512:2001, no units are specified.

The possible values of the gain may depend on the hardware. Therefore, after writing a value to this attribute, the value should be read back to know the actual value.

5.10.1.2.5 max_transmitting_gain

Holds the value of the max-transmitting-gain.

Corresponds to the maximum attenuation bound to be used by the server system in the transmitting mode. The default unit is dB.

The possible values of the gain may depend on the hardware. Therefore, after writing a value to this attribute, the value should be read back to know the actual value.

5.10.1.2.6 search_initiator_gain

This attribute is used in the intelligent search initiator process. If the value of the max_receiving_gain is below the value of this attribute, a fast synchronization process is possible.

5.10.1.2.7 frequencies

Contains frequencies required for S-FSK modulation.

```
frequencies_type::= structure
{
        mark_frequency: double-long-unsigned,
        space_frequency:        double-long-unsigned
}
```

The default unit is Hz.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.10.1.2.8 mac_address

Holds the MIB variable mac-address (variable 3) specified in

IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

NOTE 1 MAC addresses are expressed on 12 bits.

Contains the value of the address of the physical attachment (MAC address) associated to the local system. In the unconfigured state, the MAC address is “NEW-address”.

This attribute is locally written by the CIASE when the system is registered (with a Register service). The value is used in each outgoing or incoming frame. The default value is "NEW-address".

This attribute is set to NEW:

– by the MAC sub-layer, once the time-out-not-addressed delay is exceeded;
– when a client system “resets” the server system. See the “S-FSK Active initiator” IC in 4.10.4.

When this attribute is set to NEW:

– the system loses its synchronization (function of the MAC-sublayer);
– the mac_group_address attribute is reset (array of 0 elements);
– the system automatically releases all AAs which can be released.

NOTE 2 The second item is not present in IEC 61334-4-512:2001.

The predefined MAC addresses are shown in Table 33.

5.10.1.2.9 mac_group_addresses

Holds the MIB variable mac-group-address (variable 4) specified in

IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

Contains a set of MAC group addresses used for broadcast purposes.

array mac-address

mac-address::= long-unsigned

The ALL-configured-address, ALL-physical-address and NO-BODY addresses are not included in this list. These ones are internal predefined values.

This attribute shall be written by the initiator using DLMS® services to declare specific MAC group addresses on a server system.

This attribute is locally read by the MAC sublayer when checking the destination address field of a MAC frame not recognized as an individual address or as one of the three predefined values (ALL-configured-address, ALL-physical-address and NO-BODY).

5.10.1.2.10 repeater

Holds the MIB variable repeater (variable 5) specified in

IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

It specifies whether the server system effectively repeats all frames or not.

enum:

(0) never repeater,
(1) always repeater,
(2) dynamic repeater

If the repeater variable is equal to 0, the server system should never repeat the frames.

If it is set to 1, the server system is a repeater: it has to repeat all frames received without error and with a current credit greater than zero.

If it is set to 2, then the repeater status can be dynamically changed by the server itself.

NOTE The value 2 value is not specified in IEC 61334-4-512:2001.

This attribute is internally read by the MAC sub-layer each time a frame is received. The default value is 2, and the server system is a repeater (*repeater_status* = TRUE).

5.10.1.2.11 repeater_status

Holds the current repeater status of the device.

boolean:

FALSE = no repeater,
TRUE = repeater

5.10.1.2.12 min_delta_credit

Holds the MIB variable min-delta-credit (variable 9) specified in IEC 61334-4-512:2001, 5.3 and in IEC 61334-5-1:2001, 4.3.7.6.

NOTE Only the three least significant bits are used.

The Delta Credit (DC) is the subtraction of the Initial Credit (IC) and Current Credit (CC) fields of a correct received MAC frame. The delta-credit minimum value of a correct received MAC frame, directed to a server system, is held by this variable.

The default value is set to the maximal initial credit (see IEC 61334-5-1:2001, 4.2.3.1 for further explanations on the credit and the value of MAX_INITIAL_CREDIT). A client system can reinitialise this variable by setting its value to the maximal initial credit.

5.10.1.2.13 initiator_mac_address

Holds the MIB variable initiator-mac-address specified in IEC 61334-5-1:2001, 4.3.7.6.

Its value is either the MAC address of the active-initiator or the NO-BODY address, depending on the value of the synchronization_locked attribute (see 5.10.1.2.14). See also IEC 61334-5-1:2001, 3.5.3, 4.1.6.3 and 4.1.7.2.

If the value NO-BODY is written then the server mac_address (see the *mac_address* attribute) has to be set to NEW.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.10.1.2.14 synchronization_locked

Holds the MIB variable synchronization-locked (variable 10) specified in IEC 61334-4-512:2001, 5.3.

Controls the synchronization locked / unlocked state. See IEC 61334-5-1:2001 for more details.

If the value of this attribute is equal to TRUE, the system is in the synchronization-locked state. In this state, the initiator-mac-address is always equal to the MAC address field of the active-initiator MIB object. See attribute 2 of the S-FSK Active initiator IC, in 4.10.4.

If the value of this attribute is equal to FALSE, the system is in the synchronization-unlocked state. In this state, the *initiator_mac_address* attribute is always set to the NO-BODY value: a value change in the MAC address field of the active-initiator MIB object does not affect the content of the *initiator_mac_address* attribute which remains at the NO-BODY value. The default value of this variable shall be specified in the implementation specifications.

NOTE In the synchronization-unlocked state, the server synchronizes on any valid frame. In the synchronization locked state, the server only synchronizes on frames issued or directed to the client system the MAC address of which is equal to the value of the *initiator_mac_address* attribute.

5.10.2 S-FSK IEC 61334-4-32 LLC setup (class_id = 55, version = 0)

5.10.2.1 Overview

An instance of the “S-FSK IEC 61334-4-32 LLC setup” IC holds parameters necessary to set up and manage the LLC layer as specified in IEC 61334-4-32:1996.

S-FSK IEC 61334-4-32 LLC setup		**0…n**	**class_id = 55, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. max_frame_length	(static)	unsigned	26	242	134	x + 0x08
3. reply_status_list	(dyn.)	array				x + 0x10
Specific methods		***m/o***				

5.10.2.2 Attribute description

5.10.2.2.1 logical_name

Identifies the “S-FSK IEC 61334-4-32 LLC setup” object instance. See 6.2.24.

5.10.2.2.2 max_frame_length

Holds the length of the LLC frame in bytes. See IEC 61334-4-32:1996, 5.1.4.

In the case of the S-FSK profile, as specified in IEC 61334-5-1:2001, 4.2.2, the maximum value is 242, but lower values may be chosen due to performance considerations.

5.10.2.2.3 reply_status_list

Holds the MIB variable reply-status-list (variable 11) specified in IEC 61334-4-512:2001, 5.4.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Lists the L-SAPs that have a not empty RDR (Reply Data on Request) buffer, which has not already been read. The length of a waiting L-SDU is specified in number of sub frames (different from zero). The variable is locally generated by the LLC sub layer.

```
array        reply_status

reply_status::= structure
{
        L-SAP-selector:              unsigned,
        length-of-waiting-L-SDU:     unsigned
}
```

length-of-waiting-LSDU in the case of the S-FSK profile is in number of sub-frames; valid values are 1 to 7.

5.11 Previous versions of interface classes for setting up the LLC layer for ISO/IEC 8802-2

There are no previous versions to report.

5.12 Previous versions of interface classes for setting up and managing DLMS®/COSEM narrowband OFDM PLC profile for PRIME networks

There are no previous versions to report.

5.13 Previous versions of interface classes for setting up and managing DLMS®/COSEM narrowband OFDM PLC profile for G3-PLC networks

5.13.1 Mapping of G3-PLC IB attributes to COSEM IC attributes (Original version)

In terms of IEEE 802.15.4:2006, a meter is Reduced Function Device (RFD) while a concentrator / Neighbourhood Network Access Point (NNAP) is a Full Function Device (FFD) / PAN coordinator. In terms of DLMS®/COSEM the meter is the server and the concentrator / NNAP is the client (or an agent for a client).

As COSEM models only the server and not the client, the G3 NB OFDM PLC setup classes concern only the RFD (Reduced Function Device) and not the PAN coordinator.

Table 50 shows the mapping of G3 NB OFDM PLC PIB attributes to attributes of COSEM interface classes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 50 – Mapping of G3-PLC IB attributes specified in ITU-T G.9903:2017 to COSEM IC attributes

Name	Identifier	Interface class	class_id / attribute
MAC counters – Read only PIB attributes that provide statistic information [1)]			
mac_Tx_data_packet_count	0x02000101	PRIME NB OFDM PLC Physical layer parameters (class_id = 90, version = 0) (5.13.2)	90 / Att. 2
mac_Rx_data_packet_count	0x02000102		90 / Att. 3
mac_Tx_cmd_packet_count	0x02000201		90 / Att. 4
mac_Rx_cmd_packet_count	0x02000202		90 / Att. 5
mac_CSMA_fail_count	0x02000103		90 / Att. 6
mac_CSMA_no_ACK_count	0x02000104		90 / Att. 7
mac_bad_CRC_count	0x02000108		90 / Att. 8
mac_broadcast_count	0x02000106		90 / Att. 9
mac_multicast_count	0x02000107		90 / Att. 10
MAC setup PIB attributes – Read only & read-write variables [1)]			
mac_short_address	0x01000112	G3 NB OFDM PLC MAC setup (class_id = 91, version = 0) (5.13.3)	91 / Att. 2
mac_coord_short_address	0x01000107		91 / Att. 3
mac_PAN_id	0x0100010F		91 / Att. 4
mac_max_orphan_timer	0x02000109		91 / Att. 5
mac_security_enabled	0x01000111		91 / Att. 6
mac_freq_notching	0x0000006D		91 / Att. 7
mac_TMR_TTL	0x02000113		91 / Att. 8
mac_max_frame_retries	0x0100010D		91 / Att. 9
mac_neighbour_table_entry_TTL	0x02000114		91 / Att. 10
mac_neighbour_table	0x0000006B		91 / Att. 11
6LoWPAN adaptation layer IB attributes – Read only & read-write variables [2)]			
adp_max_hops	0x10	G3 NB OFDM PLC 6LoWPAN adaptation layer setup (Class_id = 92, version = 0) (5.13.4)	92 / Att. 2
adp_weak_LQI_value	0x1B		92 / Att. 3
adp_tone_mask	0x0F		92 / Att. 4
adp_discovery_attempts_wait_time	0x06		92 / Att. 5
adp_routing_configuration	0x12 – 0x19, 0x1A, 0x1C, 0x26		92 / Att. 6
adp_broadcast_log_table_entry_TTL	0x02		92 / Att. 7
adp_routing_table	0x0C		92 / Att. 8
adp_context_information_table	0x07		92 / Att. 9
adp_blacklist_table	0x25		92 / Att. 10
adp_broadcast_log_table	0x0B		92 / Att. 11
adp_group_table	0x0E		92 / Att. 12
adp_max_join_wait_time	0x21		92 / Att. 13
adp_path_discovery_time	0x22		92 / Att. 14
adp_use_new_GMK_time	0x23		92 / Att. 15
adp_exp_prec_GMK_time	0x24		92 / Att. 16

[1)] See 3GPP TS 36.321 V15.5.0 (2019-05) clauses 9.3.6.2.2 and 9.3.6.2.3

[2)] See 3GPP TS 36.321 V15.5.0 (2019-05) clause 9.4.1.1

NOTE Whereas in 3GPP TS 36.321 V15.5.0 (2019-05) the camel notation is used, in COSEM interface class specifications – and in this table – the underscore notation is used.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.2 G3 NB OFDM PLC MAC layer counters (class_id = 90, version = 0)

5.13.2.1 Overview

An instance of the “G3 NB OFDM PLC MAC layer counters” IC stores counters related to the MAC layer exchanges. The objective of these counters is to provide statistical information for management purposes.

The attributes of instances of this IC shall be read only. They can be reset using the reset method.

G3 NB OFDM PLC MAC layer counters		0...n	class_id = 90, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. mac_Tx_data_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x08
3. mac_Rx_data_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x10
4. mac_Tx_cmd_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x18
5. mac_Rx_cmd_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x20
6. mac_CSMA_fail_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x28
7. mac_CSMA_no_ACK_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x30
8. mac_bad_CRC_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x38
9. mac_broadcast_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x40
10. mac_multicast_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x48
Specific methods		***m/o***				
1. reset (data)		*o*				x + 0x50

5.13.2.2 Attribute description

5.13.2.2.1 logical_name

Identifies the “G3 NB OFDM PLC MAC layer counters” object instance. See 6.2.28.

5.13.2.2.2 mac_Tx_data_packet_count

PIB attribute 0x02000101: Statistic counter of successfully transmitted data packets (MSDUs).

5.13.2.2.3 mac_Rx_data_packet_count

PIB attribute 0x02000102: Statistic counter of successfully received data packets (MSDUs).

5.13.2.2.4 mac_Tx_cmd_packet_count

PIB attribute 0x02000201: Statistic counter of successfully transmitted command packets.

5.13.2.2.5 mac_Rx_cmd_packet_count

PIB attribute 0x02000202: Statistic counter of successfully received command packets.

5.13.2.2.6 mac_CSMA_fail_count

PIB attribute 0x02000103: Counts the number of times when CSMA backoffs exceed macMaxCSMABackoffs.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.2.2.7 mac_CSMA_no_ACK_count

PIB attribute 0x02000104: Counts the number of times when an ACK is not received while transmitting a unicast data frame (The loss of ACK is attributed to collisions).

5.13.2.2.8 mac_bad_CRC_count

PIB attribute 0x02000108: Statistic counter of the number of frames received with bad CRC.

5.13.2.2.9 mac_broadcast_count

PIB attribute 0x02000106: Statistic counter of the number of broadcast frames sent.

5.13.2.2.10 mac_multicast_count

PIB attribute 0x02000107: Statistic counter of the number of multicast frames sent.

NOTE When a counter reaches the maximum value (0xFFFFFFFF), it is automatically rolled-over.

5.13.2.3 Method description

5.13.2.3.1 reset (data)

This method forces a reset of the object. By invoking this method, the value of all counters is set to 0.

data::= integer (0)

5.13.3 G3 NB OFDM PLC MAC setup (class_id = 91, version = 0)

5.13.3.1 Overview

An instance of the “G3 NB OFDM PLC MAC setup” IC holds the necessary parameters to set up and manage the G3 NB OFDM PLC IEEE 802.15.4:2006 MAC sub-layer.

These attributes influence the functional behaviour of an implementation. Implementations may allow changes to the attributes during normal running, i.e. even after the device start-up sequence has been executed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3 NB OFDM PLC MAC setup	0...n	class_id = 91, version = 0			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				X
2. mac_short_address (dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x08
3. mac_coord_short_address (dyn.)	long-unsigned	0x0000	0xFFFF	0x0000	x + 0x10
4. mac_PAN_id (dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x18
5. mac_max_orphan_timer (static)	double-long-unsigned	0	4 294 967 295	0	x + 0x20
6. mac_security_enabled (static)	boolean			TRUE	x + 0x28
7. mac_freq_notching (static)	boolean			FALSE	x + 0x30
8. mac_TMR_TTL (static)	double-long-unsigned	0	262 143	120	x + 0x38
9. mac_max_frame_retries (static)	unsigned	0	10	5	x + 0x40
10. mac_neighbour_table_entry_TTL (static)	double-long-unsigned	0	262 143	15 300	x + 0x48
11. mac_neighbour_table (dyn.)	array				
Specific methods	***m/o***				
1. mac_get_neighbour_table_entry (data)	o				x + 0x60

5.13.3.2 Attribute description

5.13.3.2.1 logical_name

Identifies the “G3 NB OFDM PLC MAC setup” object instance. See 6.2.28.

5.13.3.2.2 mac_short_address

PIB attribute 0x01000112: Device short address.

The 16-bit address the device is using to communicate on the PAN. Its value shall be equal to 0xFFFF when the device does not have a short address. An associated device necessarily has a short address, so that a device cannot be in the state where it is associated but does not have a short address.

5.13.3.2.3 mac_coord_short_address

PIB attribute 0x01000107: Coordinator short address.

NOTE The short address assigned to the coordinator through which the device is associated.

5.13.3.2.4 mac_PAN_id

PIB attribute 0x0100010F: PAN ID.

NOTE The 16-bit identifier of the PAN on which the device is operating. A value equal to 0xFFFF indicates that the device is not associated.

5.13.3.2.5 mac_max_orphan_timer

PIB attribute 0x02000109: The maximum number of seconds without communication with a particular device after which it is declared as an orphan.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.3.2.6 mac_security_enabled

PIB attribute 0x01000111: Security enabled.

boolean:

TRUE: security enabled,
FALSE: security disabled

5.13.3.2.7 mac_freq_notching

PIB attribute 0x0000006D: S-FSK 63 and 74 kHz frequency notching.

boolean:

TRUE: notching enabled,
FALSE: notching disabled

Default value is FALSE (disabled).

5.13.3.2.8 mac_TMR_TTL

PIB attribute 0x02000113: Maximum valid time of tone map parameters in the neighbour table in seconds.

5.13.3.2.9 mac_max_frame_retries

PIB attribute 0x0100010D: Maximum number of retransmissions.

5.13.3.2.10 mac_neighbour_table_entry_TTL

PIB attribute 0x02000114: Maximum valid time for an entry in the neighbour table in seconds.

5.13.3.2.11 mac_neighbour_table

PIB attribute 0x0000006B: See ITU-T G.9903:2017, 9.3.7.2 for CENELEC and FCC bands.

The neighbour table contains information about all the devices within the POS of the device. One element of the table represents one PLC direct neighbour of the device.

```
array        mac_neighbour_table_element

mac_neighbour_table_element::= structure
{
        mac_short_address:                  long-unsigned,
        payload_modulation_scheme:              boolean,
        tone_map:                       bit-string,
        modulation:                     enum,
        tx_gain:                        integer,
        tx_res:                             boolean,
        tx_coeff:                       array,
        lqi:                            unsigned,
        TMR_valid_time:                 double-long-unsigned,
        neighbour_valid_time:               double-long-unsigned
}
```

NOTE 1 This table is actualized each time any frame is received from a neighbour device, and each time a Tone Map Response is received.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

mac_short_ address	The MAC Short Address of the node which this entry refers to.	
payload_ modulation_ scheme	0: Differential, 1: Coherent	
tone_map	The Tone Map parameter defines which frequency sub-band can be used for communication with the device. A bit set to 1 means that the frequency sub-band can be used, and a bit set to 0 means that frequency sub-band shall not be used.	
modulation	The modulation type to use for communicating with the device.	
	enum:	(0) Robust Mode (1) DBPSK (2) DQPSK (3) D8PSK (4) 16-QAM
	NOTE 2 The 16-QAM modulation is optional and only applicable for FCC band.	
tx_gain	Defines the Tx Gain to use to transmit frames to that device.	
tx_res	Defines the Tx Gain resolution corresponding to one gain step. 0: 6 dB 1: 3 dB	
tx_coeff	A parameter that specifies transmitter gain for each group of tones represented by one valid bit of the tone map. The receiver measures the frequency-dependent attenuation of the channel and may request the transmitter to compensate for this attenuation by increasing the transmit power on sections of the spectrum that are experiencing attenuation in order to equalize the received signal. Each group of tones is mapped to a 4-bit value for CENELEC-A or a 2-bit value for FCC where a "0" in the most significant bit indicates a positive gain value, hence an increase in the transmitter gain scaled by TXRES is requested for that section and a "1" indicates a negative gain value, hence a decrease in the transmitter gain scaled by TXRES is requested for that section. Implementing this feature is optional and it is intended for frequency selective channels. If this feature is not implemented, the value zero shall be used. NOTE 3 One group of tones gathers 6 consecutive tones (or carriers) for CENELEC bands, and 3 consecutive tones for FCC band.	
lqi	Link quality indicator NOTE 4 LQI value measured during reception of the PPDU from the neighbour. The LQI measurement is a characterization of the strength and/or quality of a received packet.	
TMR_valid_ time	Remaining time in seconds until which the tone map response parameters in the neighbour table are considered valid. – When the entry is created, this value shall be set to the default value 0. – When it reaches 0, a tone map request may be issued if data is sent to this device. Upon successful reception of a tone map response, this value is set to macTMRTTL.	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

neighbour_ valid_time	Remaining time in seconds until which this entry in the neighbour table is considered valid. Every time an entry is created or a frame (data or ACK) is received from this neighbour, it is set to macNeighbourTableEntryTTL. When it reaches zero, this entry is no longer valid in the table and may be removed.

5.13.3.3 Method description

5.13.3.3.1 mac_get_neighbour_table_entry (data)

This method is used to retrieve the mac neighbour table for one MAC short address. It may be used to perform topology monitoring by the client.

The method invocation parameter contains a mac_short_address.

data::= long-unsigned

The response parameter includes the neighbour table for this mac_short_address.

data::= array *mac_neighbour_table_*element

where mac_neighbour_table-element is as defined in the *mac_neighbour_table* attribute of the present IC.

5.13.4 G3 NB OFDM PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 0)

5.13.4.1 Overview

An instance of the “G3 NB OFDM PLC 6LoWPAN adaptation layer setup” IC holds the necessary parameters to set up and manage the G3 NB OFDM PLC 6LoWPAN Adaptation layer.

These attributes influence the functional behaviour of an implementation. Implementations may allow changes to their values during normal running, i.e. even after the device start-up sequence has been executed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3 NB OFDM PLC 6LoWPAN adaptation layer setup		0...n	class_id = 92, version = 0			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. adp_max_hops	(static)	unsigned	1	14	8	x + 0x10
3. adp_weak_LQI_value	(static)	unsigned	0	255	3 for CENELEC A 5 for FCC	x + 0x18
4. adp_tone_mask	(static)	bit-string			All bits set to 1	x + 0x20
5. adp_discovery_attempts_wait_time	(static)	long-unsigned	1	3 600	60	x + 0x28
6. adp_routing_configuration	(static)	array				x + 0x30
7. adp_broadcast_log_table_entry_ TTL	(static)	long-unsigned	0	3 600	90	x + 0x38
8. adp_routing_table	(dyn.)	array				x + 0x40
9. adp_context_information_table	(dyn)	array				x + 0x48
10. adp_blacklist_table	(dyn)	array				x + 0x50
11. adp_broadcast_log_table	(dyn)	array				x + 0x58
12. adp_group_table	(dyn)	array				x + 0x60
13. adp_max_join_wait_time	(static)	long-unsigned	0	1 023	20	x + 0x68
14. adp_path_discovery_time	(static)	long-unsigned	0	65535	5 000	x + 0x70
15. adp_use_new_GMK_time	(static)	long-unsigned	0	65535	3 600	x + 0x78
16. adp_exp_prec_GMK_time	(static)	long-unsigned	0	3 600	3 600	X + 0x80
Specific methods		***m/o***				

5.13.4.2 Attribute description

5.13.4.2.1 logical_name

Identifies the “G3 NB OFDM 6LoWPAN adaptation layer setup” object instance. See 6.2.28.

5.13.4.2.2 adp_max_hops

PIB attribute 0x10: Defines the maximum number of hops to be used by the routing algorithm.

5.13.4.2.3 adp_weak_LQI_value

PIB attribute 0x1B: The weak link value defines the threshold below which a direct neighbour is not taken into account during the commissioning procedure (compared to the LQI measured).

5.13.4.2.4 adp_tone_mask

PIB attribute 0x0F: Defines the Tone Mask to use during symbol formation. The length of the bit-string is 70 bits.

5.13.4.2.5 adp_discovery_attempts_wait_time

PIB attribute 0x06: Allows programming the maximum wait time between invocations of two consecutive network discovery primitives (in seconds).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.4.2.6 adp_routing_configuration

The routing configuration element specifies all parameters linked to the routing mechanism described in ITU-T G.9903:2017. The elements are specified in 9.4.1.2 of that Recommendation.

NOTE The Link cost calculation is provided in ITU-T G.9903:2017, Annex B.

```
array routing_configuration

routing_configuration::= structure
{
    adp_net_traversal_time:            long-unsigned,
    adp_routing_table_entry_TTL:       double-long-unsigned,
    adp_Kr:                            unsigned,
    adp_Km:                            unsigned,
    adp_Kc:                            unsigned,
    adp_Kq:                            unsigned,
    adp_Kh:                            unsigned,
    adp_Krt:                           unsigned,
    adp_RREQ_retries:                  unsigned,
    adp_RREQ_RERR_wait:                long-unsigned,
    adp_blacklist_table_entry_TTL:     long-unsigned
}
```

adp_net_traversal_time	PIB attribute 0x12: The Max duration between RREQ and the correspondent RREP (in seconds). Range: 0-65 535 Default value: 20
adp_routing_table_entry_TTL	PIB attribute 0x13: The time to live of a route request table entry (in seconds). Range: 0-262 143 Default value: 90
adp_Kr	PIB attribute 0x14: A weight factor for ROBO to calculate link cost. Range: 0-31 Default value: 0
adp_Km	PIB attribute 0x15: A weight factor for modulation to calculate link cost. Range: 0-31 Default value: 0
adp_Kc	PIB attribute 0x16: A weight factor for number of active tones to calculate link cost. Range: 0-31 Default value: 0
adp_Kq	PIB attribute 0x17: A weight factor for LQI to calculate route cost. Range: 0-31 Default value: 10 for CENELEC A band / 40 for FCC band.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

adp_Kh	PIB attribute 0x18: A weight factor for hop to calculate link cost. Range: 0-31 Default value: 4 for CENELEC A band / 2 for FCC band.
adp_Krt	PIB attribute 0x1C: A weight factor for the number of active routes in the routing table to calculate link cost. Range: 0-31 Default value: 0
adp_RREQ_ retries	PIB attribute 0x19: The number of RREQ re-transmission in case of RREP reception time out. Range: 0-255 Default value: 0
adp_RREQ_ RERR_wait	PIB attribute 0x1A: The number of seconds to wait between two consecutive RREQ – RERR generations. Range: 0-65 535 Default value: 30
adp_ blacklist_ table_entry_ TTL	PIB attribute 0x26: Time to live of a blacklisted neighbour set entry in minutes. Range: 0-65 535 Default value: 10

5.13.4.2.7 adp_broadcast_log_table_entry_TTL

PIB attribute 0x02: Defines the time while an entry in the adpBroadcastLogTable remains active in the table (in seconds).

5.13.4.2.8 adp_routing_table

PIB attribute 0x0C: The routing table contains information about the different routes in which the device is implicated.

```
array routing_table

routing_table::= structure
{
destination_address:    long-unsigned,
next_hop_address:       long-unsigned,
route_cost:             long-unsigned,
hop_count:              unsigned,
weak_link_count:        unsigned,
status:                 enum,
valid_time:             unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

destination_ address	Address of the destination.	
next_hop_ address	Address of the next hop on the route towards the destination.	
route_cost	Cumulative link cost along the route towards the destination.	
hop_count	Number of hops of the selected route to the destination. Range: 0-15 NOTE 1 Practically the maximum allowed value is limited by adp_max_hops.	
weak_link_ count	Number of weak links to destination. Range : 0-15	
NOTE 2 Practically the maximum allowed value is limited by adp_max_hops.		
status	Status of the routing table entry.	
	enum:	(0) Invalid route, (1) Valid route, (2) Route under construction
valid_time	Remaining time in seconds until which this entry in the routing table is considered valid.	

5.13.4.2.9 adp_context_information_table

PIB attribute 0x07: Contains the context information associated to each CID extension field.

```
array        context_information_table
context_information_table::= structure
{
        CID:                bit-string,
        context_length:     unsigned,
        context:            octet-string,
        C:                  boolean,
        valid_lifetime:     long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

CID	Corresponds to the 4-bit context information used for source and destination addresses (SCI, DCI). Range: 0x00-0x0F
context_ length	Indicates the length of the carried context (up to 128-bit contexts may be carried). Range: 0-128
context	Corresponds to the carried context used for compression/decompression purposes. Range: 0x0000:0000:0000:0000:0000:0000:0000:0000 – 0xFFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
C	Indicates if the context is valid for use in compression 0: Only decompression is allowed, 1: Compression and decompression are allowed
A context may be used for decompression purposes only. Moreover, recommendations made in RFC 6775 should be followed to take into account the propagation of the context to all nodes of the PAN.	
valid_ lifetime	Remaining time in minutes during which the context information table is considered valid. It is updated upon reception of the advertised context. Range: 0-65 535

5.13.4.2.10 adp_blacklist_table

PIB attribute 0x25: Contains the list of the blacklisted neighbours.

```
array        blacklisted_neighbour_set
blacklisted_neighbour_set::= structure
{
        blacklisted_neighbour_address:    long-unsigned,
        valid_time:                       long-unsigned
}
```

blacklisted_ neighbour_ address	The 16-bit address of the blacklisted neighbour.
valid_time	Remaining time in minutes until which this entry in the blacklisted neighbour table is considered valid.

5.13.4.2.11 adp_broadcast_log_table

PIB attribute 0x0B: Contains the broadcast log table.

NOTE This table provides a list of the broadcast packets recently received by this device.

```
array        broadcast_log_table
broadcast_log_table::= structure
{
        source_address:        long-unsigned,
        sequence_number:       unsigned,
        time_to_live:          long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

source_ address	The 16-bit source address of a broadcast packet. This is the address of the broadcast initiator.
sequence_ number	The sequence number contained in the BC0 header.
time_to_ live	The remaining time to live of this entry in the broadcast log table, in seconds.

5.13.4.2.12 adp_group_table

PIB attribute 0x0E: Contains the group addresses to which the device belongs.

```
array        group_table
group_table::= structure
{
        group_address:   long-unsigned
}
```

group_address Group address to which this node has been subscribed.

5.13.4.2.13 adp_max_join_wait_time

PIB attribute 0x21: Network join timeout in seconds for LBD.

5.13.4.2.14 adp_path_discovery_time

PIB attribute 0x22: Timeout for path discovery in msec.

5.13.4.2.15 adp_use_new_GMK_time

PIB attribute 0x23: The wait time in seconds for a device to use new GMK after rekeying.

5.13.4.2.16 adp_exp_prec_GMK_time

PIB attribute 0x24: The time in seconds to keep PrecGMK after switching to a new GMK.

5.13.5 Mapping of G3-PLC PIB attributes to COSEM IC attributes (previous version)

In terms of IEEE 802.15.4:2006, a meter is a Reduced Function Device (RFD) while a concentrator / Neighbourhood Network Access Point (NNAP) is a Full Function Device (FFD) / PAN coordinator. In terms of DLMS/COSEM the meter is the server and the concentrator / NNAP is the client (or an agent for a client).

As COSEM models only the server and not the client, the G3-PLC setup classes concern only the RFD (Reduced Function Device) and not the PAN coordinator.

Table 51 shows the mapping of G3-PLC PIB attributes to attributes of COSEM interface classes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table 51 – Mapping of G3-PLC IB attributes to COSEM IC attributes

Name	Identifier	Interface class		class_id / attribute
MAC counters – Read only PIB attributes that provide statistic information [1]				
mac_Tx_data_packet_count	0x0101	G3-PLC MAC layer counters (class_id = 90, version = 1) (4.13.3)		90 / Att. 2
mac_Rx_data_packet_count	0x0102			90 / Att. 3
mac_Tx_cmd_packet_count	0x0103			90 / Att. 4
mac_Rx_cmd_packet_count	0x0104			90 / Att. 5
mac_CSMA_fail_count	0x0105			90 / Att. 6
mac_CSMA_no_ACK_count	0x0106			90 / Att. 7
mac_bad_CRC_count	0x0109			90 / Att. 8
mac_Tx_data_broadcast_count	0x0108			90 / Att. 9
mac_Rx_data_broadcast_count	0x0107			90 / Att. 10
MAC setup PIB attributes – Read only & read-write & write only variables [1), 2)]				
mac_short_address	0x0053	G3-PLC MAC setup (class_id = 91, version = 1) (5.13.6)		91 / Att. 2
mac_RC_coord	0x010F			91 / Att. 3
mac_PAN_id	0x0050			91 / Att. 4
mac_key_table	0x0071			91 / Att. 5
mac_frame_counter	0x0077			91 / Att. 6
mac_tone_mask	0x0110			91 / Att. 7
mac_TMR_TTL	0x010D			91 / Att. 8
mac_max_frame_retries	0x0059			91 / Att. 9
mac_neighbour_table_entry_TTL	0x010E			91 / Att. 10
mac_neighbour_table	0x010A			91 / Att. 11
mac_high_priority_window_size	0x0100			91 / Att. 12
mac_CSMA_fairness_limit	0x010C			91 / Att. 13
mac_beacon_randomization_window_length	0x0111			91 / Att. 14
mac_A	0x0112			91 / Att. 15
mac_K	0x0113			91 / Att. 16
mac_min_CW_attempts	0x0114			91 / Att. 17
mac_cenelec_legacy_mode	0x0115			91 / Att. 18
mac_FCC_legacy_mode	0x0116			91 / Att. 19
mac_max_BE	0x0047			91 / Att. 20
mac_max_CSMA_backoff	0x004E			91 / Att. 21
mac_min_BE	0x004F			91 / Att. 22
6LoWPAN adaptation layer IB attributes – Read only & read-write variables [3), 4)]				
adp_max_hops	0x0F	G3-PLC MAC setup (class_id = 91, version = 1) (5.13.6)		92 / Att. 2
adp_weak_LQI_value	0x1A			92 / Att. 3
adp_security_level	0x00			92 / Att. 4
adp_prefix_table	0x01			92 / Att. 5
adp_routing_configuration	0x09, 0x0A, 0x0D, 0x11-0x19, 0x1B, 0x1F			92 / Att. 6
adp_broadcast_log_table_entry_TTL	0x02			92 / Att. 7
adp_routing_table	0x0C			92 / Att. 8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Name	Identifier	Interface class		class_id / attribute
adp_context_information_table	0x07			92 / Att. 9
adp_blacklist_table	0x1E			92 / Att. 10
adp_broadcast_log_table	0x0B			92 / Att. 11
adp_group_table	0x0E			92 / Att. 12
adp_max_join_wait_time	0x20			92 / Att. 13
adp_path_discovery_time	0x21			92 / Att. 14
adp_active_key_index	0x22			92 / Att. 15
adp_metric_type	0x03			92 / Att. 16
adp_coord_short_address	0x08			92 / Att. 17
adp_disable_default_routing	0xF0			92 / Att. 18
adp_device_type	0x10			92 / Att. 19

[1] See ITU-T G.9903:2014, 9.3.6.2.2 and 9.3.6.2.3.

[2] The following attributes of the G3-PLC MAC sublayer IB attributes have been excluded as there is no need to expose them: *macBSN*, *macDSN*, *macAckWaitDuration*, *macFreqNotching*, *macTimeStampSupported*, *macPromiscuousMode*, *macSecurityEnabled*.

[3] See ITU-T G.9903:2014, 9.4.1.1.

[4] The following attributes of the G3-PLC Adaptation sublayer IB attributes have been excluded as there is no need to expose them; *adpSoftVersion, adpSnifferMode*.

NOTE Whereas in 3GPP TS 36.321 V15.5.0 (2019-05) the camel-case notation is used, in COSEM interface class specifications – and in this table – the underscore notation is used.

5.13.6 G3-PLC MAC setup (class_id = 91, version = 1)

5.13.6.1 Overview

An instance of the “G3-PLC MAC setup” IC holds the necessary parameters to set up and manage the G3-PLC IEEE 802.15.4 MAC sub-layer.

These attributes influence the functional behaviour of an implementation. Implementations may allow changes to the attributes during normal running, i.e. even after the device start-up sequence has been executed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3-PLC MAC setup		0...n	class_id = 91, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
23. logical_name	(static)	octet-string				x
24. mac_short_address	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x08
25. mac_RC_coord	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x10
26. mac_PAN_id	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x18
27. mac_key_table	(dyn.)	array				x + 0x20
28. mac_frame_counter	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x28
29. mac_tone_mask	(static)	bit-string			0x00000000FFFFFFFF	x + 0x30
30. mac_TMR_TTL	(static)	unsigned	0	255	2	x + 0x38
31. mac_max_frame_retries	(static)	unsigned	0	10	5	x + 0x40
32. mac_neighbour_table_entry_TTL	(static)	unsigned	0	255	255	x + 0x48
33. mac_neighbour_table	(dyn.)	array				x + 0x50
34. mac_high_priority_window_size	(static)	unsigned	1	7	7	x + 0x58
35. mac_CSMA_fairness_limit	(static)	unsigned	See below	255	25	x + 0x60
36. mac_beacon_randomization_window_length	(static)	unsigned	1	254	12	x + 0x68
37. mac_A	(static)	unsigned	3	20	8	x + 0x70
38. mac_K	(static)	unsigned	1	See below	5	x + 0x78
39. mac_min_CW_attempts	(static)	unsigned	0	255	10	x + 0x80
40. mac_cenelec_legacy_mode	(static)	unsigned	0	255	1	x + 0x88
41. mac_FCC_legacy_mode	(static)	unsigned	0	255	1	x + 0x90
42. mac_max_BE	(static)	unsigned	0	20	8	x + 0x98
43. mac_max_CSMA_backoff	(static)	unsigned	0	255	50	x + 0xA0
44. mac_min_BE	(static)	unsigned	0	20	3	x + 0xA8
Specific methods		***m/o***				
2. mac_get_neighbour_table_entry (data)		o				x + 0xB0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.6.2 Attribute description

5.13.6.2.1 logical_name

Identifies the “G3-PLC MAC setup” object instance. See 6.2.28.

5.13.6.2.2 mac_short_address

PIB attribute 0x0053: The 16-bit address the device is using to communicate through the PAN. Its value shall be equal to 0xFFFF when the device does not have a short address. An associated device necessarily has a short address, so that a device cannot be in the state where it is associated but does not have a short address.

5.13.6.2.3 mac_RC_coord

PIB attribute 0x010F: Route cost to coordinator, to be used in the beacon payload as RC_COORD.

5.13.6.2.4 mac_PAN_id

PIB attribute 0x0050: The 16-bit identifier of the PAN through which the device is operating. A value equal to 0xFFFF indicates that the device is not associated.

5.13.6.2.5 mac_key_table

PIB attribute 0x0071: This attribute holds GMK keys required for MAC layer ciphering. The attribute can hold up to two 16-bytes keys. The Key Identifier value shall be different for each key.

For security reason, the key entries cannot be read, only written.

```
array mac_GMK
mac_GMK::= structure
{
        key_id:      unsigned,
        key:   octet-string
}
```

Where:

key_id The Key Identifier used to refer to this key, can take the value 0 or 1.

key The AES-128 key used for ciphering the frames exchanged at MAC layer.

5.13.6.2.6 mac_frame_counter

PIB attribute 0x0077: The outgoing frame counter for this device, used when ciphering frames at MAC layer.

5.13.6.2.7 mac_tone_mask

PIB attribute 0x0110: Defines the tone mask to use during symbol formation.

5.13.6.2.8 mac_TMR_TTL

PIB attribute 0x010D: Maximum time to live of tone map parameters entry in the neighbour table in minutes.

5.13.6.2.9 mac_max_frame_retries

PIB attribute 0x0059: Maximum number of retransmissions.

5.13.6.2.10 mac_neighbour_table_entry_TTL

PIB attribute 0x010E: Maximum time to live for an entry in the neighbour table in minutes.

5.13.6.2.11 mac_neighbour_table

PIB attribute 0x010A: See ITU-T G.9903:2017, 9.3.7.2 for CENELEC and FCC bands.

The neighbour table contains information about all the devices within the POS of the device. One element of the table represents one PLC direct neighbour of the device.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
array neighbour_table
neighbour_table::= structure
{
    short_address:                long-unsigned,
    payload_modulation_scheme:    boolean,
    tone_map:                     bit-string,
    modulation:                   enum,
    tx_gain:                      integer,
    tx_res:                       enum,
    tx_coeff:                     bit-string,
    lqi:                          unsigned,
    phase_differential            integer,
    TMR_valid_time:               unsigned,
    neighbour_valid_time:         unsigned
}
```

NOTE 1 This table is actualized each time any frame is received from a neighbour device, and each time a Tone Map Response is received.

Where:

short_ address	The MAC Short Address of the node which this entry refers to.
payload_ modulation_ scheme	Payload Modulation scheme to be used when transmitting to this neighbour. FALSE: Differential, TRUE: Coherent
tone_map	The Tone Map parameter defines which frequency sub-band can be used for communication with the device. A bit set to 1 means that the frequency sub-band can be used, and a bit set to 0 means that frequency sub-band shall not be used.
modulation	The modulation type to use for communicating with the device. enum: (0) Robust Mode, (1) DBPSK, (2) DQPSK, (3) D8PSK, (4) 16-QAM NOTE 2 The 16-QAM modulation is optional and only applicable for FCC band.
tx_gain	Defines the Tx Gain to use to transmit frames to that device.
tx_res	Defines the Tx Gain resolution corresponding to one gain step. 0: 6 dB, 1: 3 dB

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

tx_coeff	A parameter that specifies transmitter gain for each group of tones represented by one valid bit of the tone map. The receiver measures the frequency-dependent attenuation of the channel and may request the transmitter to compensate for this attenuation by increasing the transmit power on sections of the spectrum that are experiencing attenuation in order to equalize the received signal. Each group of tones is mapped to a 4-bit value for CENELEC-A or a 2-bit value for FCC where a "0" in the most significant bit indicates a positive gain value, hence an increase in the transmitter gain scaled by TXRES is requested for that section and a "1" indicates a negative gain value, hence a decrease in the transmitter gain scaled by TXRES is requested for that section. Implementing this feature is optional and it is intended for frequency selective channels. If this feature is not implemented, the value zero shall be used. *Example: in CENELEC bands, with gain values equal to [1, -5, 4, -2, 0, 1], tx_coeff encoding is equal to: '0001 1101 0100 1010 0000 0001'* NOTE 3 One group of tones gathers 6 consecutive tones (or carriers) for CENELEC bands, and 3 consecutive tones for FCC band.
lqi	Link Quality Indicator of the link to the neighbour (reverse LQI). NOTE 4 LQI value measured during reception of the PPDU from the neighbour. The LQI measurement is a characterization of the strength and/or quality of a received packet.
phase_ differential	Phase difference in multiples of 60 degrees between the mains phase of the local node and the neighbour node. PhaseDifferential can assume six integer values between 0 and 5.
TMR_valid_ time	Remaining time in minutes until which the tone map response parameters in the neighbour table are considered valid. – When the entry is created, this value shall be set to the default value 0. – When it reaches 0, a tone map request may be issued if data is sent to this device. Upon successful reception of a tone map response, this value is set to mac_TMR_TTL.
neighbour_ valid_time	Remaining time in minutes until which this entry in the neighbour table is considered valid. Every time an entry is created or a frame (data or ACK) is received from this neighbour, it is set to mac_neighbour_table_entry_TTL. When it reaches zero, this entry is no longer valid in the table and may be removed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.6.2.12 mac_high_priority_window_size

PIB attribute 0x0100: The high priority contention window size in number of slots.

5.13.6.2.13 mac_CSMA_fairness_limit

PIB attribute 0x010C: Channel access fairness limit. Specifies how many failed back-off attempts, back-off exponent is set to minBE. This attribute can take a value between 2 × (macMaxBE – macMinBE) and 255.

5.13.6.2.14 mac_beacon_randomization_window_length

PIB attribute 0x0111: Duration time in seconds for the beacon randomization.

5.13.6.2.15 mac_A

PIB attribute 0x0112: This parameter controls the adaptive CW linear decrease.

5.13.6.2.16 mac_K

PIB attribute 0x0113: Rate adaptation factor for channel access fairness limit. This attribute can take a value between 1 and macCSMAFairnessLimit.

5.13.6.2.17 mac_min_CW_attempts

PIB attribute 0x0114: Number of consecutive attempts while using minimum CW.

5.13.6.2.18 mac_cenelec_legacy_mode

PIB attribute 0x0115: This read only attribute indicates the capability of the node.

0: The following configuration is used (legacy mode):

- Elementary interleaving;
- Interleaver parameters n_i and n_j are not swapped when I(i,j) = 0.

1: The following configuration is used (non legacy mode):

- Full Block interleaving;
- Interleaver parameters n_i and n_j are swapped when I(i,j) = 0.

5.13.6.2.19 mac_FCC_legacy_mode

PIB attribute 0x0116: This read only attribute indicates the capability of the node.

0: The following configuration is used (legacy mode):

- Differential FCH modulation;
- Elementary interleaving;
- Interleaver parameters n_i and n_j are not swapped when I(i,j) = 0;
- Single RS block.

1: The following configuration is used (non legacy mode):

- Coherent FCH modulation;
- Full Block interleaving;
- Interleaver parameters n_i and n_j are swapped when I(i,j) = 0;
- Two RS blocks.

5.13.6.2.20 mac_max_BE

PIB attribute 0x0047: Maximum value of backoff exponent. It should always be greater than macMinBE.

5.13.6.2.21 mac_max_CSMA_backoff

PIB attribute 0x004E: Maximum number of backoff attempts.

5.13.6.2.22 mac_min_BE

PIB attribute 0x004F: Minimum value of backoff exponent.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.6.3 Method description

5.13.6.3.1 mac_get_neighbour_table_entry (data)

This method is used to retrieve the mac neighbour table for one MAC short address. It may be used to perform topology monitoring by the client.

The method invocation parameter contains a mac_short_address.

data::= long-unsigned

The response parameter includes the neighbour table for this mac_short_address.

data::= array neighbour_table

where mac_neighbour_table is as defined in the *mac_neighbour_table* attribute of the present IC.

5.13.7 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 1)

5.13.7.1 Overview

An instance of the “G3-PLC 6LoWPAN adaptation layer setup” IC holds the necessary parameters to set up and manage the G3-PLC 6LoWPAN Adaptation layer.

These attributes influence the functional behaviour of an implementation. Implementations may allow changes to their values during normal running, i.e. even after the device start-up sequence has been executed.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3-PLC 6LoWPAN adaptation layer setup		0...n	class_id = 92, version = 1			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				X
2. adp_max_hops	(static)	unsigned	1	14	8	x + 0x08
3. adp_weak_LQI_value	(static)	unsigned	0	255	52	x + 0x10
4. adp_security_level	(static)	unsigned	0	7	5	x + 0x18
5. adp_prefix_table	(dyn.)	array				x + 0x20
6. adp_routing_configuration	(static)	array				x + 0x28
7. adp_broadcast_log_table_entry_TTL	(static)	long-unsigned	0	65535	2	x + 0x30
8. adp_routing_table	(dyn.)	array				x + 0x38
9. adp_context_information_table	(dyn)	array				x + 0x40
10. adp_blacklist_table	(dyn)	array				x + 0x48
11. adp_broadcast_log_table	(dyn)	array				x + 0x50
12. adp_group_table	(dyn)	array				x + 0x58
13. adp_max_join_wait_time	(static)	long-unsigned	0	1 023	20	x + 0x60
14. adp_path_discovery_time	(static)	unsigned	0	255	40	x + 0x68
15. adp_active_key_index	(static)	unsigned	0	1	0	x + 0x70
16. adp_metric_type	(static)	unsigned	0x00	0x0F	0x0F	x + 0x78
17. adp_coord_short_address	(static)	long-unsigned	0x0000	0x7FFF	0x0000	x + 0x80
18. adp_disable_default_routing	(static)	boolean			FALSE	x + 0x88
19. adp_device_type	(static)	enum	0	2	2	x + 0x90
Specific methods		***m/o***				

5.13.7.2 Attribute description

5.13.7.2.1 logical_name

Identifies the “G3-PLC OFDM 6LoWPAN adaptation layer setup” object instance. See 6.2.28.

5.13.7.2.2 adp_max_hops

PIB attribute 0x0F: Defines the maximum number of hops to be used by the routing algorithm.

5.13.7.2.3 adp_weak_LQI_value

PIB attribute 0x1A: The weak link value defines the LQI value below which a link to a neighbour is considered as a weak link. A value of 52 represents an SNR of 3 dB.

5.13.7.2.4 adp_security_level

PIB attribute 0x00: The minimum security level to be used for incoming and outgoing adaptation frames. Only values 0 (no ciphering) and 5 (ciphering with 32 bits integrity code) are supported.

5.13.7.2.5 adp_prefix_table

PIB attribute 0x01: Contains the list of prefixes defined on this PAN.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.7.2.6 adp_routing_configuration

The routing configuration element specifies all parameters linked to the routing mechanism described in ITU-T G.9903 Amd. 1:2017. The elements are specified in 9.4.1.2 of that Recommendation.

NOTE The Link cost calculation is provided in ITU-T G.9903:2017, Annex B.

```
array routing_configuration
routing_configuration::= structure
{
    adp_net_traversal_time:            unsigned,
    adp_routing_table_entry_TTL:       long-unsigned,
    adp_Kr:                            unsigned,
    adp_Km:                            unsigned,
    adp_Kc:                            unsigned,
    adp_Kq:                            unsigned,
    adp_Kh:                            unsigned,
    adp_Krt:                           unsigned,
    adp_RREQ_retries:                  unsigned,
    adp_RREQ_RERR_wait:                unsigned,
    adp_blacklist_table_entry_TTL:     long-unsigned,
    adp_unicast_RREQ_gen_enable:       boolean,
    adp_RLC_Enabled:                   boolean,
    adp_add_rev_link_cost:             unsigned
}
```

Where

adp_net_traversal_time — PIB attribute 0x11: Maximum time that a packet is expected to take to reach any node from any node in seconds.
Range : 0-255
Default value : 20

adp_routing_table_entry_TTL — PIB attribute 0x12: Maximum time-to-live of a routing table entry (in minutes).
Range : 0-65 535
Default value : 60

adp_Kr — PIB attribute 0x13: A weight factor for the Robust Mode to calculate link cost.
Range : 0-31
Default value : 0

adp_Km — PIB attribute 0x14: A weight factor for modulation to calculate link cost.
Range : 0-31
Default value : 0

adp_Kc — PIB attribute 0x15: A weight factor for number of active tones to calculate link cost.
Range : 0-31
Default value : 0

adp_Kq — PIB attribute 0x16: A weight factor for LQI to calculate route cost.
Range : 0-50
Default value : 10 for CENELEC A band / 40 for FCC band.

adp_Kh — PIB attribute 0x17: A weight factor for hop to calculate link cost.
Range : 0-31
Default value : 4 for CENELEC A band / 2 for FCC band.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

adp_Krt	PIB attribute 0x1B: A weight factor for the number of active routes in the routing table to calculate link cost. Range : 0-31 Default value : 0
adp_RREQ_retries	PIB attribute 0x18: The number of RREQ re-transmission in case of RREP reception time out. Range : 0-255 Default value : 0
adp_RREQ_RERR_wait	PIB attribute 0x19: The number of seconds to wait between two consecutive RREQ – RERR generations. Range : 0-255 Default value : 30
adp_blacklist_table_entry_TTL	PIB attribute 0x1F: Maximum time-to-live of a blacklisted neighbour entry (in minutes). Range : 0-65 535 Default value : 10
adp_unicast_RREQ_gen_enable	PIB attribute 0x0D: If TRUE, the RREQ shall be generated with its "unicast RREQ" flag set to '1'. If FALSE, the RREQ shall be generated with its "unicast RREQ" flag set to '0'. Default value : TRUE
adp_RLC_enabled	PIB attribute 0x09: Enable the sending of RLCREQ frame by the device. Default value : FALSE
adp_add_rev_ink_cost	PIB attribute 0x0A: It represents an additional cost to take into account a possible asymmetry in the link. Range : 0-255 Default value : 0

5.13.7.2.7 adp_broadcast_log_table_entry_TTL

PIB attribute 0x02: Maximum time to live of an adpBroadcastLogTable entry (in minutes).

5.13.7.2.8 adp_routing_table

PIB attribute 0x0C: Contains the routing table.

```
array routing_table

routing_table::= structure

{
destination_address:    long-unsigned,
next_hop_address:           long-unsigned,
route_cost:             long-unsigned,
hop_count:              unsigned,
weak_link_count:        unsigned,
valid_time:             long-unsigned
}
```

NOTE 1 This table is actualized each time a route is built or updated (triggered by data traffic) and each time the TTL timer expires.

Where

destination_address	Address of the destination.
next_hop_address	Address of the next hop on the route towards the destination.
route_cost	Cumulative link cost along the route towards the destination.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

hop_count — Number of hops of the selected route to the destination.
Range: 0-14

NOTE 2 Practically the maximum allowed value is limited by adp_max_hops.

weak_link_count — Number of weak links to destination.
Range: 0-14

NOTE 3 Practically the maximum allowed value is limited by adp_max_hops.

valid_time — Remaining time in minutes until when this entry in the routing table is considered valid.

5.13.7.2.9 adp_context_information_table

PIB attribute 0x07: Contains the context information associated to each CID extension field.

```
array context_information_table

context_information_table::= structure

{
CID:                bit-string,
context_length:     unsigned,
context:                octet-string,
C:                      boolean,
valid_lifetime:         long-unsigned
}
```

Where

CID — Corresponds to the 4-bit context information used for source and destination addresses (SCI, DCI).
Range: 0x00-0x0F

context_length — Indicates the length of the carried context (up to 128-bit contexts may be carried).
Range: 0-128

context — Corresponds to the carried context used for compression/decompression purposes.
Range: 0x0000:0000:0000:0000:0000:0000:0000:0000 – 0xFFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF

C — Indicates if the context is valid for use in compression.

FALSE: Only decompression is allowed,

TRUE: Compression and decompression are allowed

A context may be used for decompression purposes only. Moreover, recommendations made in RFC 6775 should be followed to take into account the propagation of the context to all nodes of the PAN.

valid_lifetime — Remaining time in minutes during which the context information table is considered valid. It is updated upon reception of the advertised context.

Range: 0-65 535

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.7.2.10 adp_blacklist_table

PIB attribute 0x1E: Contains the list of the blacklisted neighbours.

```
array blacklisted_neighbour_set

blacklisted_neighbour_set::= structure

{
    blacklisted_neighbour_address:    long-unsigned,
    valid_time:                       long-unsigned
}
```

Where

blacklisted_neighbour_address	The 16-bit address of the blacklisted neighbour.
valid_time	Remaining time in minutes until which this entry in the blacklisted neighbour table is considered valid.

5.13.7.2.11 adp_broadcast_log_table

PIB attribute 0x0B: Contains the broadcast log table.

NOTE This table provides a list of the broadcast packets recently received by this device.

```
array broadcast_log_table

broadcast_log_table::= structure

{
    source_address:       long-unsigned,
    sequence_number:      unsigned,
    valid_time:           long-unsigned
}
```

Where:

source_address	The 16-bit source address of a broadcast packet. This is the addressof the broadcast initiator;
sequence_number	The sequence number contained in the BC0 header;
valid_time	Remaining time in minutes until when this entry in the broadcast log table is considered valid.

5.13.7.2.12 adp_group_table

PIB attribute 0x0E: Contains the group addresses to which the device belongs.

```
array group_address

group_address::= long-unsigned
```

group_address Group address to which this node has been subscribed.

5.13.7.2.13 adp_max_join_wait_time

PIB attribute 0x20: Network join timeout in seconds for LBD.

5.13.7.2.14 adp_path_discovery_time

PIB attribute 0x21: Timeout for path discovery in seconds.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.13.7.2.15 adp_active_key_index

PIB attribute 0x22: Index of the active GMK to be used for data transmission.

5.13.7.2.16 adp_metric_type

PIB attribute 0x03: Metric Type to be used for routing purposes.

5.13.7.2.17 adp_coord_short_address

PIB attribute 0x08: Defines the short address of the coordinator.

5.13.7.2.18 adp_disable_default_routing

PIB attribute 0xF0: If TRUE, the default routing (LOADng) is disabled. If FALSE, the default routing (LOADng) is enabled.

5.13.7.2.19 adp_device_type

PIB attribute 0x10: Defines the type of the device connected to the modem:

enum:

(0) PAN device,
(1) PAN coordinator,
(2) Not Defined

5.14 Previous versions of interface classes for setting up and managing DLMS®/COSEM HS-PLC ISO/IEC 12139-1 neighbourhood networks

There are no previous versions to report.

5.15 Previous versions of ZigBee® setup classes

There are no previous versions to report.

6 Relation to OBIS

6.1 General

This Clause 6 specifies the use of COSEM interface objects to model various data items.

It also specifies the logical names of the objects. The naming system is based on OBIS, the Object Identification System: each logical name is an OBIS code.

OBIS codes are specified in the following subclauses:

- 6.2 specifies the use and the logical names of abstract COSEM objects, i.e. objects not related to an energy type;
- 6.3 specifies the use and logical names for electricity related COSEM objects;
- the detailed OBIS code allocations are specified in IEC 62056-6-1:2023.

  NOTE The use and the logical names of COSEM objects related to other media / energy types are under consideration.
- the detailed OBIS code allocations – for all media / energy types – are specified in IEC 62056-6-1:2023.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Unless otherwise specified the use of value group B shall be:

- if just one object is instantiated, the value in value group B shall be 0;
- if more than one object is instantiated in the same physical device, the value group B shall number the measurement or communication channels as appropriate, from 1…64. This is indicated by the letter "*b*".

Unless otherwise specified the use of value group E shall be:

- if just one object is instantiated, value in value group E shall be 0;
- if more than one object is instantiated in the same physical device, the value group E shall number the instantiations from zero to the maximum value needed. This is indicated by the letter "*e*". For the values allocated, see IEC 62056-6-1:2023 .

All codes, which are not explicitly listed, but which are outside the manufacturer, utility or consortia specific ranges are reserved for future use.

6.2 Abstract COSEM objects

6.2.1 Use of value group C

Table 52 shows the use of value group C for abstract objects in the COSEM context. See also IEC 62056-6-1:2023, Table 5.

Table 52 – Use of value group C for abstract objects in the COSEM context

Value group C Abstract objects (A = 0)	
0	General purpose COSEM objects
1	Instances of IC "Clock"
2	Instances of IC "Modem configuration" and related IC-s
10	Instances of IC "Script table"
11	Instances of IC "Special days table"
12	Instances of IC "Schedule"
13	Instances of IC "Activity calendar"
14	Instances of IC "Register activation"
15	Instances of IC "Single action schedule"
16	Instances of IC "Register monitor", "Parameter monitor"
17	Instances of IC "Limiter"
18	Instances of IC "Array manager"
19	COSEM objects related to payment metering: "Account", "Credit", "Charge", "Token gateway"
20	Instances of IC "IEC local port setup"
21	Standard readout definitions
22	Instances of IC "IEC HDLC setup"
23	Instances of IC "IEC twisted pair (1) setup"
24	COSEM objects related to M-Bus
25	Instances of IC "TCP-UDP setup", "IPv4 setup", "IPv6 setup", "MAC address setup", "PPP setup", "GPRS modem setup", "SMTP setup", "GSM diagnostic", "FTP setup", "Push setup", "NTP setup", "LTE monitoring" "SCHC-LPWAN setup", "SCHC-LPWAN diagnostic", "SCHC-LoRaWAN setup", "SCHC-LoRaWAN diagnostic"
26	COSEM objects for data exchange using S-FSK PLC networks

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Value group C Abstract objects (A = 0)	
27	COSEM objects for ISO/IEC 8802-2 LLC layer setup
28	COSEM objects for data exchange using narrow-band OFDM PLC for PRIME networks
29	COSEM objects for data exchange using narrow-band OFDM PLC for G3-PLC networks
30	COSEM objects for data exchange using ZigBee®
31	Instances of IC “Wireless Mode Q” (M-Bus)
32	Instances of IC “ISO/IEC 14908 identification”, “ISO/IEC 14908 protocol setup”, “ ISO/IEC 14908 protocol setup”, “ ISO/IEC protocol status”, “ISO/IEC 14908 diagnostic”
33	COSEM objects for data exchange using HS-PLC ISO/IEC 12139-1 networks
34	Instances of IC “Wi-SUN setup”, “Wi-SUN diagnostic”, “RPL diagnostic”, “MPL diagnostic”
40	Instances of IC “Association SN/LN”
41	Instances of IC “SAP assignment”
42	COSEM logical device name
43	COSEM objects related to security: Instances of IC “Security setup” and “Data protection”
44	Instances of IC “Image transfer” and “Function control” objects
65	Instances of IC “Utility tables”
66	Instances of “Compact data”
128…199	Manufacturer specific COSEM related abstract objects
All other	Reserved

6.2.2 Data of historical billing periods

COSEM provides three mechanisms to represent values of historical billing periods. This is shown in Figure 31 and is described below:

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

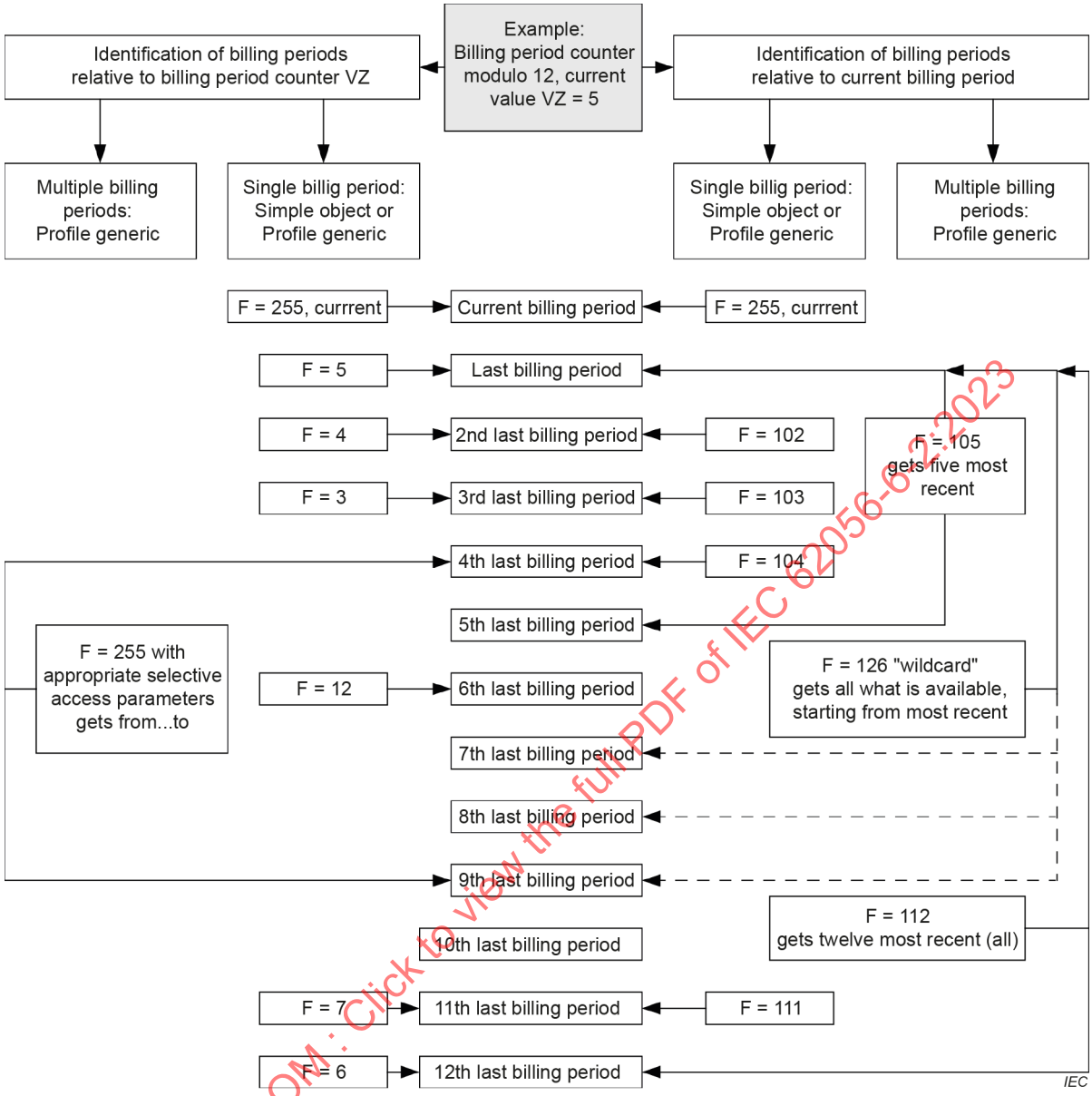


Figure 31 – Data of historical billing periods – example with module 12, VZ = 5

- a value of a single historical billing period may be represented using the same IC as used for representing the value of the current billing period. With F = 0…99, the billing period is identified by the value of the billing period counter VZ. F = VZ identifies the youngest value, F = VZ-1 identifies the second youngest value, etc. F = 101…125 identifies the last, second last …25th last billing period. (F = 255 identifies the current billing period). Simple objects can only be used to represent values of historical billing periods, if “Profile generic” objects are not implemented;
- a value of a single historical billing period may also be represented by “Profile generic” objects, which are one entry deep, and contain the historical value itself and the time stamp of the storage. With F = 0…99, the billing period is identified by the value of the billing period counter VZ. F = VZ identifies the youngest value, F = VZ-1 identifies the second youngest value etc. F=101 identifies the most recent billing period;
- values of multiple historical billing periods are represented with “Profile generic” objects, with suitable controlling attributes. With F = 102…125 the two last, …25 last values can be reached. F = 126 identifies an unspecified number of historical values;
- when values of historical billing periods are represented by “Profile generic” objects, more than one billing periods schemes may be used. The billing period scheme is identified by the billing period counter object captured in the profile.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.3 Billing period values / reset counter entries

These values are represented by instances of the IC "Data".

For billing period / reset counters and for number of available billing periods the data type shall be *unsigned*, *long-unsigned* or *double-long-unsigned*. For time stamps of billing periods, the data type shall be *double-long-unsigned* (in the case of UNIX time), *octet-string* or *date-time* formatted as specified in 4.1.6.1.

These objects may be related to energy type – see also 6.3.3 and IEC 62056-6-1:2023, Table 20 – and channels.

When the values of historical periods are represented by “Profile generic” objects, the time stamp of the billing period objects shall be part of the captured objects.

Billing period values / reset counter entries	IC	OBIS code					
		A	B	C	D	E	F
For item names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data [a]	0	*b*	0	1	*e*	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.2.4 Other abstract general purpose OBIS codes

Program entries shall be represented by instances of the IC "Data" with data type *unsigned, long-unsigned* or *octet-string*.

For identifying the firmware the following objects are available:

- Active firmware identifier objects hold the identifier of the currently active firmware;
- Active firmware version objects hold the version of the currently active firmware;

  NOTE *Firmware version* can be used to distinguish between different releases of a firmware identified by the same *Firmware identifier*.
- Active firmware signature objects hold the digital signature of the currently active firmware. The digital signature algorithm is not specified here.

These three elements are inextricably linked to the currently active firmware.

Firmware identifiers may be also energy type and channel related.

Time entry values shall be represented by instances of IC “Data”, “Register” or “Extended register” with the data type of the value attribute octet-string, formatted as *date-time* in 4.1.6.1.

For detailed OBIS codes, see IEC 62056-6-1:2023, Table 8.

Abstract general purpose OBIS codes	IC	OBIS code					
		A	B	C	D	E	F
Program entries	1, Data [a]	0	*b*	0	2	*e*	255
Time entries		0	*b*	0	9	*e*	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.5 Clock objects (class_id = 8)

Instances of the IC "Clock" – see 4.5.1 – control the system clock of the physical device.

"UNIX clock" objects are instances of the Interface class "Data", with data type *double-long-unsigned*. They hold the number of seconds since 1970-01-01 00:00:00.

NOTE 1 DLMS® uses *double-long-unsigned* rather than the UNIX standard *double-long-signed* data type to extend the roll over date to 2106.

"Microseconds clock" objects are instances of the IC "Data", with data type *long64-unsigned*. They hold the number of microseconds since 1970-01-01 00:00:00.

NOTE 2 Previously, the name of these objects was "High resolution clock".

"Minutes clock" objects are instances of the IC "Data", with data type *double-long-unsigned*. They hold the whole number of minutes since 1970-01-01 00:00:00 UTC.

"Hours clock" objects are instances of the IC "Data", with data type *double-long-unsigned*. They hold the whole number of the hours since 1970-01-01 00:00:00 UTC.

"Days clock" objects are instances of the IC "Data", with data type *long-unsigned*. They hold the whole number of the days since 1970-01-01 00:00:00 UTC.

"Weeks clock" objects are instances of the IC "Data", with data type *long-unsigned*. They hold the whole number of weeks since 1970-01-01 00:00:00 UTC.

These objects are all inter-related in that it is anticipated that there is a single notion of time within the device.

Clock objects	IC	OBIS code					
		A	B	C	D	E	F
Clock	8, Clock	0	*b*	1	0	*e*	255
UNIX clock	1, Data [a]	0	*b*	1	1	*e*	255
Microseconds clock	1, Data [a]	0	*b*	1	2	*e*	255
Minutes clock	1, Data [a]	0	*b*	1	3	*e*	255
Hours clock	1, Data [a]	0	*b*	1	4	*e*	255
Days clock	1, Data [a]	0	*b*	1	5	*e*	255
Weeks clock	1, Data [a]	0	*b*	1	6	*e*	255
[a]							

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.6 Modem configuration and related objects

In this group, the following objects are available:

- Instances of the IC "Modem configuration" – see 4.7.4 – define and control the behaviour of the device regarding the communication through a modem;
- Instances of the IC "Auto connect" – see 4.7.6 – define the necessary parameters for the management of sending information from the metering device to one or more destinations and for connection to the network;
- Instances of the IC "Auto answer" – see 4.7.5 – define and control the behaviour of the device regarding the auto answering function using a modem and handling wake-up calls and messages.

Modem configuration and related objects	IC	OBIS code					
		A	B	C	D	E	F
Modem configuration	27, Modem configuration	0	*b*	2	0	0	255
Auto connect	29, Auto connect	0	*b*	2	1	0	255
Auto answer	28, Auto answer	0	*b*	2	2	0	255

6.2.7 Script table objects (class_id = 9)

Instances of the IC “Script table” – see 4.5.2 – control the behaviour of the device.

Several instances are predefined and normally available as hidden scripts only with access to the *execute* () method. The following table contains only the identifiers for the “standard” instances of the listed scripts. Implementation specific instances of these scripts should use values different from zero in value group D.

- *MDI reset / End of billing period* “Script table” objects define the actions to be performed at the end of the billing period, for example the reset of maximum demand indicator registers and archiving data. If there are several billing period schemes available, then there shall be one script present in the array of scripts for each billing period scheme;
- *Tariffication* “Script table” objects define the entry point into tariffication by standardizing utility-wide how to invoke the activation of certain tariff conditions;
- *Disconnect control* “Script table” objects hold the scripts to invoke the methods of “Disconnect control” objects;
- *Image activation* “Script table” objects are is used to locally activate an Image transferred to the server, at the date and time held by an Image activation “Single action schedule” object;
- *Push* “Script table” objects hold scripts to activate the push operation. Normally every entry in the array of scripts calls the push method of one “Push setup” object instance;
- *Load profile control* “Script table” allow to change attributes of “Profile generic” objects e.g. to change the capture period and thus allow extended time control;
- *M-Bus profile control* “Script table” allow to change attributes of M-Bus related “Profile generic” objects e.g. to change the capture period and thus allow extended time control;
- *Function control* “Script table” objects allow making changes to “Function control” objects;
- *Broadcast* “Script table” objects allow standardising utility wide the entry point into regularly needed functionality.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Script table objects	IC	OBIS code					
		A	B	C	D	E	F
Global meter reset [a] Script table	9, Script table	0	*b*	10	0	0	255
MDI reset / End of billing period [a] Script table		0	*b*	10	0	1	255
Tariffication Script table		0	*b*	10	0	100	255
Activate test mode [a] Script table		0	*b*	10	0	101	255
Activate normal mode [a]Script table		0	*b*	10	0	102	255
Set output signals Script table		0	*b*	10	0	103	255
Switch optical test output [b, c] Script table		0	*b*	10	0	104	255
Power quality measurement management Script table		0	*b*	10	0	105	255
Disconnect control Script table		0	*b*	10	0	106	255
Image activation Script table		0	*b*	10	0	107	255
Push Script table		0	*b*	10	0	108	255
Load profile control Script table		0	*b*	10	0	109	255
M-Bus profile control Script table		0	*b*	10	0	110	255
Function control Script table		0	*b*	10	0	111	255
Broadcast Script table		0	*b*	10	0	125	255

[a] The activation of these scripts is performed by calling the execute() method to the script identifier 1 of the corresponding script object.

[b] The optical test output is switched to measuring quantity Y and the test mode is activated by calling the execute method of the script table object 0.x.10.0.104.255 using Y as parameter; where Y is given by see IEC 62056-6-1:2023, Table 13. The default value of A is 1 (Electricity).

EXAMPLE In the case of electricity meters, A = 1, default, execute (21) switches the test output to display the active power + of phase 1.

[c] The optical test output is also switched back to its default value when this script is activated.

6.2.8 Special days table objects (class_id = 11)

Instances of the IC “Special days table” – see 4.5.4 – define and control the behaviour of the device regarding calendar functions on special days for clock control.

Special days table objects	IC	OBIS code					
		A	B	C	D	E	F
Special days table	11, Special days table	0	*b*	11	0	*e*	255

6.2.9 Schedule objects (class_id = 10)

Instances of the IC “Schedule” – see 4.5.3 – define and control the behaviour of the device in a sequenced way.

Schedule objects	IC	OBIS code					
		A	B	C	D	E	F
Schedule	10, Schedule	0	*b*	12	0	*e*	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.10 Activity calendar objects (class_id = 20)

Instances of the IC “Activity calendar” – see 4.5.5 – define and control the behaviour of the device in a calendar-based way.

Activity calendar objects	IC	OBIS code					
		A	B	C	D	E	F
Activity calendar	20, Activity calendar	0	*b*	13	0	*e*	255

6.2.11 Register activation objects (class_id = 6)

Instances of the IC “Register activation”– see 4.3.5 – are used to handle different tariffication structures.

Register activation objects	IC	OBIS code					
		A	B	C	D	E	F
Register activation	6, Register activation	0	*b*	14	0	*e*	255

6.2.12 Single action schedule objects (class_id = 22)

Instances of the IC “Single action schedule” – see 4.5.7 – control the behaviour of the device. Implementation specific instances should use values different from zero in value group D.

Instances of Push “Single action schedule” objects activate scripts in Push “Script table” objects, which invoke the push method of the appropriate “Push setup” objects.

Load profile control “Single action schedule” objects activate scripts in Load profile control “Script table” objects and thus allow extended time control.

M-Bus profile control “Single action schedule” objects activate scripts in M-Bus profile control “Script table” objects and thus allow extended time control.

Function control “Single action schedule” objects activate scripts in Function control “Script table” objects.

Single action schedule objects	IC	OBIS code					
		A	B	C	D	E	F
End of billing period Single action schedule	22, Single action schedule	0	*b*	15	0	0	255
Disconnect control Single action schedule		0	*b*	15	0	1	255
Image activation Single action schedule		0	*b*	15	0	2	255
Output control Single action schedule		0	*b*	15	0	3	255
Push Single action schedule		0	*b*	15	0	4	255
Load profile control Single action schedule		0	*b*	15	0	5	255
M-Bus profile control Single action schedule		0	*b*	15	0	6	255
Function control Single action schedule		0	*b*	15	0	7	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.13 Register monitor objects (class_id = 21)

Instances of the IC “Register monitor” – see 4.5.6 – control the registerand alarm monitoring function of the device. They define the value to be monitored, the set of thresholds to which the value is compared, and the actions to be performed when a threshold is crossed.

In general, the logical name(s) shown in the table below shall be used. See also 6.3.9 and 6.3.10.

Alarm monitor objects monitor *Alarm register* or *Alarm descriptor* objects.

Register monitor objects	IC	OBIS code					
		A	B	C	D	E	F
Register monitor	21, Register monitor	0	*b*	16	0	*e*	255
Alarm monitor		0	*b*	16	1	0…9	255

6.2.14 Parameter monitor objects (class_id = 65)

Instances of the IC “Parameter monitor” – see 5.5.1 – control the Parameter monitoring function of the device. They define the list of parameters to be monitored and hold the identifier and the value of the last parameter changed, as well as the *capture_time*.

Parameter monitor objects	IC	OBIS code					
		A	B	C	D	E	F
Parameter monitor	65, Parameter monitor	0	*b*	16	2	*e*	255

6.2.15 Limiter objects (class_id = 71)

Instances of the IC “Limiter” handle the monitoring of values in normal and emergency conditions. See also 4.5.9.

Limiter objects	IC	OBIS code					
		A	B	C	D	E	F
Limiter	71, Limiter	0	*b*	17	0	*e*	255

6.2.16 Array manager objects (class_id = 123)

Instances of the IC “Array manager” – see 4.4.11 – allow managing COSEM interface object attributes of type *array*.

Array manager objects	IC	OBIS code					
		A	B	C	D	E	F
Array manager	123	0	*b*	18	0	*e*	255

6.2.17 Payment metering related objects

Payment accounting can be applied to any commodity.

An instance of the “Account” – see 4.6.2 – IC holds the summary information for a given contract and lists the “Credit” and “Charge” objects used by that “Account”. If more than one “Account” is for any reason required in a given context then field D should be other than 0.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

One or several instances of the “Credit” IC – see 4.6.3 – represent the different credit sources.

One or several instances of the “Charge” IC – see 4.6.4 – represent the different charges applicable.

One or more instances of the “Token gateway” IC – see 4.6.5 – are available to enter tokens. If only a single gateway is defined in a single “Account” then field E of the OBIS code shall be zero. If more than one “Token gateway” object is for any reason required in a single “Account” then field E should be other than 0.

The “Account” is linked to it’s associated “Credit”, “Charge” and “Token gateway” objects by use of the value group D and B field such that an “Account” with D=0 should be linked to a “Token gateway” with D=40 and have a “Credit” objects with D=10 and “Charge” objects with D=20. Whereas an “Account” with D=1 should have “token gateway” with D=41, “Credit” objects with D=11 and “Charge” objects with D=21 etc. Multiple “Credit” and “Charge” objects are identified using different values in the value group E field. See also Additional Notes there describing the “Max credit_limit” and „Max vend limit” objects there.

Instances of “Profile Generic“ IC hold the history of the token credit and of the charge collections with a “Parameter Monitor” interface class monitoring a value used to trigger capture.

Payment metering related objects	IC	OBIS code					
		A	B	C	D	E	F
Account	111, Account	0	*b*	19	0..9	0	255
Credit	112, Credit	0	*b*	19	10..19	e	255
Charge	113, Charge	0	*b*	19	20..29	e	255
Token gateway	115, Token gateway	0	*b*	19	40…49	*e*	255
Configurable limit objects							
Max credit limit	01, Data	0	*b*	19	50…59	1	255
Max vend limit	01, Data	0	*b*	19	50…59	2	255

6.2.18 IEC local port setup objects (class_id = 19)

These objects define and control the behaviour of local ports using the protocol specified in IEC 62056-21:2002. See also 4.7.1.

IEC local port setup objects	IC	OBIS code					
		A	B	C	D	E	F
IEC optical port setup	19, IEC local port setup	0	*b*	20	0	0	255
IEC electrical port setup		0	*b*	20	0	1	255

6.2.19 Standard readout profile objects (class_id = 7)

A set of objects is defined to carry the standard readout as it would appear with IEC 62056-21:2002 (modes A to D). See also 4.3.6.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Standard readout objects	IC	OBIS code					
		A	B	C	D	E	F
General local port readout	7, Profile generic	0	*b*	21	0	0	255
General display readout		0	*b*	21	0	1	255
Alternate display readout		0	*b*	21	0	2	255
Service display readout		0	*b*	21	0	3	255
List of configurable meter data		0	*b*	21	0	4	255
Additional readout profile 1		0	*b*	21	0	5	255
.....							
Additional readout profile *n*		0	*b*	21	0	N	255

For the parametrization of the standard readout “Data” objects can be used.

Standard readout parametrization objects	IC	OBIS code					
		A	B	C	D	E	F
Standard readout parametrization	1, Data	0	*b*	21	0	*e*	255

6.2.20 IEC HDLC setup objects (class_id = 23)

Instances of the IC “IEC HDLC setup” – see 4.7.2 – hold the parameters of the HDLC based data link layer.

IEC HDLC setup objects	IC	OBIS code					
		A	B	C	D	E	F
IEC HDLC setup	23, IEC HDLC setup	0	*b*	22	0	0	255

6.2.21 IEC twisted pair (1) setup objects (class_id = 24)

An instance of the IC “IEC twisted pair (1) set up” IC – see 4.7.3 – stores the parameters necessary to manage a communication profile specified in IEC 62056-3-1:2013.

An instance of the IC “MAC address set up” IC stores the Secondary Station Address ADS.

An instance of the “Data” stores the Fatal Error register.

Instances of the IC “Profile generic” IC instances allow the configuration of IEC 62056-3-1 readout lists.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC twisted pair (1) setup and related objects	IC	OBIS code					
		A	B	C	D	E	F
IEC twisted pair (1) setup	24, IEC twisted pair (1) setup	0	*b*	23	0	0	255
IEC twisted pair (1) MAC address setup	43, MAC address setup	0	*b*	23	1	0	255
IEC twisted pair (1) Fatal Error register	1, Data	0	b	23	2	0	255
IEC 62056-3-1 Short readout	7, Profile generic	0	*b*	23	3	0	255
IEC 62056-3-1 Long readout		0	*b*	23	3	1	255
IEC 62056-3-1 Alternate readout profile 0		0	b	23	3	2	255
IEC 62056-3-1 Additional readout profile 1		0	b	23	3	3	255
IEC 62056-3-1 Additional readout profile 2		0	b	23	3	4	255
IEC 62056-3-1 Additional readout profile 7		0	b	23	3	9	255

For the parametrization of the IEC 62056-3-1 readout “Data” objects can be used.

Standard readout parametrization objects	IC	OBIS code					
		A	B	C	D	E	F
IEC 62056-3-1 readout parametrization	1, Data	0	*b*	23	3	*e*	255

6.2.22 Objects related to data exchange over M-Bus

The following objects are available to model and control data exchange using the M-Bus protocol specified in the EN 13757 series:

- instances of the IC “M-Bus slave port setup” define and control the behaviour of M-Bus slave ports of a DLMS®/COSEM device. See 4.8.2;
- instances of the IC “M-Bus client” are used to configure DLMS®/COSEM devices as M-Bus clients. There is one “M-Bus client” object for each M-Bus slave. Value group B identifies the M-Bus channels. See 4.8.3;
- M-Bus value objects, instances of the IC “Extended register”, hold the values captured from M-Bus slave devices on the relevant channel. The link between the M-Bus client setup objects and the M-Bus value objects is provided by the channel number;
- M-Bus “Profile generic” objects capture M-Bus value objects possibly along with other, not M-Bus specific objects;
- M-Bus “Disconnect control” objects control disconnect devices of M-Bus devices (e.g. gas valves);
- instances of the IC “Wireless mode Q” define and control the behaviour of the device regarding the communication parameters according to mode Q of EN 13757-5:2015. A node having more than one network address, i.e. a multi-homed node, will have multiple objects of these types. See 4.8.4;
- M-Bus control log objects are instances of the IC “Profile generic”. They log the changes of the state of the disconnect devices;
- instances of the IC “M-Bus master port setup” define and control the behaviour of M-Bus master ports of DLMS®/COSEM devices, allowing to exchange data with M-Bus slaves. See 4.8.5;
- instances of the IC ”DLMS®/COSEM server M-Bus port setup” are used in DLMS®/COSEM servers hosted by M-Bus slave devices, using the DLMS®/COSEM wired or wireless M-Bus communication profile. See 4.8.6;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- instances of the IC “M-Bus diagnostic” hold information related to the operation of the M-Bus network. See 4.8.7.

Objects related to data exchange over M-Bus	IC	OBIS code					
		A	B	C	D	E	F
M-Bus slave port setup	25, M-Bus slave port setup	0	*b*	24	0	0	255
M-Bus client	72, M-Bus client	0	*b*	24	1	0	255
M-Bus value	4, Extended register	0	*b*	24	2	e [a]	255
M-Bus profile generic	7, Profile generic	0	*b*	24	3	e	255
M-Bus disconnect control	70, Disconnect control	0	*b*	24	4	0	255
M-Bus control log	7, Profile generic	0	*b*	24	5	0	255
M-Bus master port setup	74, M-Bus master port setup	0	*b*	24	6	0	255
Wireless Mode Q channel	73, Wireless Mode Q channel	0	*b*	31	0	0	255
DLMS®/COSEM server M-Bus port setup	76, DLMS®/COSEM server M-Bus port setup	0	*b*	24	8	e [b]	255
M-Bus diagnostic	77, M-Bus diagnostic	0	*b*	24	9	e [b]	255

[a] “e” is equal to the index of the captured value in accordance to index of capture_definition_element in the *capture_definition* attribute of the M-Bus client object.

[b] If there is more than one M-Bus network interface present then there may be one object instantiated for each interface. For example, if a device has two interfaces (one wired M-Bus and one wireless M-Bus) and uses the DLMS®/COSEM M-Bus communication profiles on both, there shall be one instance for each interface.

6.2.23 Objects to set up data exchange over the Internet

In this group, the following objects are available:

- Instances of the IC “TCP-UDP setup” – see 4.9.1 – handle all information related to the setup of the TCP and UDP layer of the Internet based communication profile(s), and point to the IP setup object(s) handling the setup of the IP layer on which the TCP-UDP connection(s) is (are) used;
- Instances of the IC “IPv4 setup” – see 4.9.2 – handle all information related to the setup of the IPv4 layer of the Internet based communication profile(s) and point to the data link layer setup object(s) handling the setup of the data link layer on which the IP connections is (are) used;
- Instances of the IC “IPv6 setup” – see 4.9.3 – handle all information related to the setup of the IPv6 layer of the Internet based communication profile(s) and point to the data link layer setup object(s) handling the setup of the data link layer on which the IP connections is (are) used;
- Instances of the IC “MAC address setup” – see 4.9.4. – handle all information related to the setup of the Ethernet data link layer of the Internet based communication profile(s);
- Instances of the IC “PPP setup” – see 4.9.5 – handle all information related to the setup of the PPP data link layer of the Internet based communication profiles;
- Instances of the IC “GPRS modem setup” – see 4.7.7 – handle all information related to the setup of the GPRS modem;
- Instances of the IC “SMTP setup” – see 4.9.6 – handle all information related to the setup of the SMTP service;

NOTE The following objects have internet related OBIS codes, although they are not strictly related to use over the internet only.

- Instances of the IC “GSM diagnostic” – see 5.7.9 – handle all diagnostic information related to the GSM/GPRS network;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- Instances of the IC “Push setup” – see 4.4.8 – handle all information about the data to be pushed, the push destination and the method the data should be pushed;
- Instances of the IC "NTP setup" – 4.9.7 – see handle all information related to the setup of the NTP time synchronisation service;
- Instances of the IC “LTE monitoring” – see 4.7.9 – allow monitoring LTE modems;
- Instances of the IC “SCHC-LPWAN setup” – see 4.16.2.1 – handle all information related to the setup of the SCHC-LPWAN layer;
- Instances of the IC “SCHC-LPWAN diagnostic” – see 4.16.2.2 – handle all diagnostic information related to the SCHC-LPWAN network.

Objects to set up data exchange over the Internet	IC	OBIS code					
		A	B	C	D	E	F
TCP-UDP setup	41, TCP-UDP setup	0	*b*	25	0	0	255
IPv4 setup	42, IPv4 setup	0	*b*	25	1	0	255
MAC address setup	43, MAC address setup	0	*b*	25	2	0	255
PPP setup	44, PPP setup	0	*b*	25	3	0	255
GPRS modem setup	45, GPRS modem setup	0	*b*	25	4	0	255
SMTP setup	46, SMTP setup	0	*b*	25	5	0	255
GSM diagnostic	47, GSM diagnostic	0	*b*	25	6	0	255
IPv6 setup	48, IPv6 setup	0	*b*	25	7	0	255
Reserved for FTP setup							
Push setup	40, Push setup	0	*b*	25	9	0	255
NTP setup	100, NTP setup	0	*b*	25	10	0	255
LTE monitoring	151, LTE monitoring	0	*b*	25	11	0	255
SCHC-LPWAN setup	126, SCHC-LPWAN setup	0	*b*	25	12	0	255
SCHC-LPWAN diagnostic	127, SCHC-LPWAN diagnostic	0	*b*	25	13	0	255
SCHC-LoRaWAN setup	128, SCHC – LoRaWAN setup	0	*b*	25	14	0	255
SCHC-LoRaWAN diagnostic	129, SCHC – LoRaWAN diagnostic	0	*b*	25	15	0	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.24 Objects to set up Push Setup (class_id = 40)

Instances of the IC “Push setup” – see 4.4.8 – handle all information about the data to be pushed, the push destination and the method by which the data should be pushed.

Push Setup	IC	OBIS code					
		A	B	C	D	E	F
Push setup	40, Push setup	0	*b*	25	9	0	255

6.2.25 Objects for setting up data exchange using S-FSK PLC

In this group, the following objects are available:

- Instances of the IC "S-FSK Phy&MAC setup" – see 4.10.3 – handle all information related to setting up the PLC S-FSK lower layer profile specified in IEC 61334-5-1:2001.
- Instances of the IC "S-FSK Active initiator" – see 4.10.4 – handle all information related to the active initiator in the PLC S-FSK lower layer profile specified in IEC 61334-5-1:2001.
- Instances of the IC "S-FSK MAC synchronization timeouts" – see 4.10.5 – manage all timeouts related to the synchronization process of devices using the PLC S-FSK lower layer profile specified in IEC 61334-5-1:2001.
- Instances of the IC "S-FSK MAC counters" – see 4.10.6 – store counters related to the frame exchange, transmission and repetition phases in the PLC S-FSK lower layer profile specified in IEC 61334-5-1:2001.
- Instances of the IC "IEC 61334-4-32 LLC setup" – see 4.10.7 – handle all information related to the LLC layer specified in IEC 61334-4-32:1996.
- Instances of the IC "S-FSK Reporting system list" – see 4.10.8 – hold information on reporting systems in the PLC S-FSK lower layer profile specified in IEC 61334-5-1:2001.

Objects to set up data exchange using S-FSK PLC	IC	OBIS code					
		A	B	C	D	E	F
S-FSK Phy&MAC setup	50, S-FSK Phy&MAC setup	0	*b*	26	0	0	255
S-FSK Active initiator	51, S-FSK Active initiator	0	*b*	26	1	0	255
S-FSK MAC synchronization timeouts	52, S-FSK MAC synchronization	0	*b*	26	2	0	255
S-FSK MAC counters	53, S-FSK MAC counters	0	*b*	26	3	0	255
NOTE This is a placeholder for a Monitoring IC to be specified.							
IEC 61334-4-32 LLC setup	55, IEC 61334-4-32 LLC setup	0	*b*	26	5	0	255
S-FSK Reporting system list	56, S-FSK Reporting system list	0	*b*	26	6	0	255

6.2.26 Objects for setting up the ISO/IEC 8802-2 LLC layer

In this group, the following objects are available:

- Instances of the IC "ISO/IEC 8802-2 LLC Type 1 setup" – see 4.11.2 – handle all information related to the LLC layer specified in ISO/IEC 8802-2:1998 in Type 1 operation.
- Instances of the IC "ISO/IEC 8802-2 LLC Type 2 setup" – see 4.11.3 – handle all information related to the LLC layer specified in ISO/IEC 8802-2:1998 in Type 2 operation.
- Instances of the IC "ISO/IEC 8802-2 LLC Type 3 setup" – see 4.11.4 – handle all information related to the LLC layer specified in ISO/IEC 8802-2:1998 in Type 3 operation.

Objects to set up the ISO/IEC 8802-2 LLC layer	IC	OBIS code					
		A	B	C	D	E	F
ISO/IEC 8802-2 LLC Type 1 setup	57, ISO/IEC 8802-2 LLC Type 1 setup	0	*b*	27	0	0	255
ISO/IEC 8802-2 LLC Type 2 setup	58, ISO/IEC 8802-2 LLC Type 2 setup	0	*b*	27	1	0	255
ISO/IEC 8802-2 LLC Type 3 setup	59, ISO/IEC 8802-2 LLC Type 3 setup	0	*b*	27	2	0	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.27 Objects for data exchange using narrowband OFDM PLC for PRIME networks

For setting up and managing data exchange using narrowband OFDM PLC for PRIME networks one instance of each following classes shall be implemented for each interface:

- an instance of the 61334-4-32 LLC SSCS setup – see 4.12.3 – holds the addresses related to the CL_432 layer;
- an instance of the IC "PRIME NB OFDM PLC Physical layer counters" – see 4.12.5 – stores counters related to the physical layers exchanges;
- an instance of the IC "PRIME NB OFDM PLC MAC setup" – see 4.12.6 – holds the necessary parameters to set up the PRIME NB OFDM PLC MAC layer;
- an instance of the IC "PRIME NB OFDM PLC MAC functional parameters" – see 4.12.7 – provides information on specific aspects concerning the functional behaviour of the MAC layer;
- an instance of the IC "PRIME NB OFDM PLC MAC counters" – see 4.12.8 – stores statistical information on the operation of the MAC layer for management purposes;
- an instance of the IC "PRIME NB OFDM PLC MAC network administration data" – see 4.12.9 – holds the parameters related to the management of the devices connected to the network;
- an instance of the IC "MAC address setup" – holds the MAC address of the device. See 4.12.10;
- an instance of the IC "PRIME NB OFDM PLC Application identification" – see 4.12.11 – holds identification information related to administration and maintenance of PRIME NB OFDM PLC devices.

Objects for data exchange using PRIME NB OFDM PLC	IC	OBIS code					
		A	B	C	D	E	F
61334-4-32 LLC SSCS setup	80, 61334-4-32 LLC SSCS setup	0	*b*	28	0	0	255
PRIME NB OFDM PLC Physical layer counters	81, PRIME NB OFDM PLC Physical layer counters	0	*b*	28	1	0	255
PRIME NB OFDM PLC MAC setup	82, PRIME NB OFDM PLC MAC setup	0	*b*	28	2	0	255
PRIME NB OFDM PLC MAC functional parameters	83, PRIME NB OFDM PLC MAC functional parameters	0	*b*	28	3	0	255
PRIME NB OFDM PLC MAC counters	84, PRIME NB OFDM PLC MAC counters	0	*b*	28	4	0	255
PRIME NB OFDM PLC MAC network administration data	85, PRIME NB OFDM PLC MAC network administration data	0	*b*	28	5	0	255
PRIME NB OFDM PLC MAC address setup	43, MAC address setup	0	*b*	28	6	0	255
PRIME NB OFDM PLC Application identification	86, PRIME NB OFDM PLC Application identification	0	*b*	28	7	0	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.28 Objects for data exchange using narrow-band OFDM PLC for G3-PLC networks

For setting up and managing data exchange using G3-PLC profile, one instance of each following classes shall be implemented for each interface:

- an instance of the IC "G3-PLC MAC layer counters" – see 4.13.3 – to store counters related to the MAC layer exchanges;
- an instance of the IC "G3-PLC MAC setup" – see 4.13.4 – to hold the necessary parameters to set up the G3-PLC MAC IEEE 802.15.4:2006 layer;

- an instance of the IC “G3-PLC 6LoWPAN adaptation layer setup” – see 4.13.5 – to hold the necessary parameters to set up the Adaptation layer.

Objects for data exchange using G3-PLC	IC	OBIS code					
		A	B	C	D	E	F
G3-PLC MAC layer counters	90, G3-PLC MAC layers counters	0	*b*	29	0	0	255
G3-PLC MAC setup	91, G3-PLC MAC setup	0	*b*	29	1	0	255
G3-PLC 6LoWPAN adaptation layer setup	92, G3-PLC 6LoWPAN adaptation layer setup	0	*b*	29	2	0	255

6.2.29 ZigBee® setup objects

The following objects are available for setting up and managing a ZigBee® network, see also 4.15.

Objects set up and manage ZigBee® networks	IC	OBIS code					
		A	B	C	D	E	F
ZigBee® SAS startup	101, ZigBee® SAS Startup	0	*b*	30	0	e	255
ZigBee® SAS join	102, ZigBee® SAS join	0	*b*	30	1	e	255
ZigBee® SAS APS fragmentation	103, ZigBee® SAS APS fragmentation	0	*b*	30	2	e	255
ZigBee® network control	104, ZigBee® network control	0	*b*	30	3	e	255
ZigBee® tunnel setup	105, ZigBee® tunnel setup	0	*b*	30	4	e	255

6.2.30 Objects for setting up and managing data exchange using ISO/IEC 14908 PLC networks

For setting up and managing data exchange using ISO/IEC 14908 PLC networks at least one instance of each of the following interface classes shall be implemented, see also 4.16

Objects to set up and manage ISO/IEC 14908 PLC netowrks	IC	OBIS identification					
		A	B	C	D	E	F
ISO/IEC 14908 identification	130, ISO/IEC 14908 identification	0	*b*	32	0	0	255
ISO/IEC 14908 protocol setup	131, ISO/IEC 14908 protocol setup	0	*b*	32	1	0	255
ISO/IEC 14908 protocol status	132, ISO/IEC 14908 protocol status	0	*b*	32	2	0	255
ISO/IEC 14908 diagnostic	133, ISO/IEC 14908 diagnostic	0	*b*	32	3	0	255

6.2.31 Objects for data exchange using HS-PLC ISO/IEC 12139-1 ISO/EC 12139-1 networks

For setting up and managing data exchange using HS-PLC ISO/IEC 12139-1 networks one instance of each following classes shall be implemented for each interface:

- an instance of the IC “HS-PLC ISO/IEC 12139-1 MAC setup” – see 4.14.2 – holds the necessary parameters for setting up the MAC layer;
- an instance of the IC “HS-PLC ISO/IEC 12139-1 CPAS setup” – see 4.14.3 – holds the necessary parameters for setting up the CPAS;
- an instance of the IC “HS-PLC ISO/IEC 12139-1 IP SSAS setup” – see 4.14.4 – holds the necessary parameters for setting up the IP SSAS;
- an instance of the IC “HS-PLC ISO/IEC 12139-1 HDLC SSAS setup” – see 4.14.5 – holds the necessary parameters for setting up the HDLC SSAS.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Objects for data exchange using HS-PLC ISO/IEC 12139-1	IC	OBIS code					
		A	B	C	D	E	F
HS-PLC ISO/IEC 12139-1 MAC setup	140, HS-PLC ISO/IEC 12139-1 MAC setup	0	*b*	33	0	0	255
HS-PLC ISO/IEC 12139-1 CPAS setup	141, HS-PLC ISO/IEC 12139-1 CPAS setup	0	*b*	33	1	0	255
HS-PLC ISO/IEC 12139-1 IP SSAS setup	142, HS-PLC ISO/IEC 12139-1 IP SSAS setup	0	*b*	33	2	0	255
HS-PLC ISO/IEC 12139-1 HDLC SSAS setup	143, HS-PLC ISO/IEC 12139-1 HDLC SSAS setup	0	*b*	33	3	0	255

6.2.32 Objects for data exchange using Wi-SUN networks

For setting up and managing data exchange using Wi-SUN networks one instance of each following classes shall be implemented for each interface:

- An instance of the IC “IPv6 setup” – see 4.9.3 – handle all information related to the setup of the IPv6 layer of the Internet based communication profile(s) and point to the data link layer setup object(s) handling the setup of the data link layer on which the IP connections is (are) used.
- An instance of the IC “Wi-SUN setup” – see 4.17.1 – handle all information related to the setup of the SMTP service.
- An instance of the IC “Interface diagnostic” – see 4.17.2 – handle all diagnostic information related to the network.
- An instance of the IC “RPL diagnostic” – see 4.17.3 – handle all diagnostic information related to RPL.
- An instance of the IC “MPL diagnostic” – see 4.17.4 – handle all diagnostic information related to MPL.

Objects to set up data exchange over the Wi-SUN FAN	IC	OBIS code					
		A	B	C	D	E	F
Wi-SUN setup	95, Wi-SUN setup	0	*b*	34	0	0	255
Wi-SUN diagnostic	96, Wi-SUN diagnostic	0	*b*	34	1	0	255
RPL diagnostic	97, RPL diagnostic	0	*b*	34	2	0	255
MPL diagnostic	98, MPL diagnostic	0	*b*	34	3	0	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.33 Association objects (class_id = 12, 15)

A series of Association SN / LN objects – see 4.4.3, 4.4.4 – are available to model application associations between a DLMS®/COSEM client and server.

Association objects	IC	OBIS code					
		A	B	C	D	E	F
Current association	12, Association SN 15, Association LN	0	0	40	0	0	255
Association, instance 1		0	0	40	0	1	255
……							
Association, instance *n*		0	0	40	0	n	255

6.2.34 SAP assignment object (class_id = 17)

An instance of the IC "SAP assignment" – see 4.4.5 – holds information about the addresses (Service Access Points, SAPs) of logical devices within a physical device.

SAP Assignment object	IC	OBIS code					
		A	B	C	D	E	F
SAP assignment of current physical device	17, SAP assignment	0	0	41	0	0	255

6.2.35 COSEM logical device name object

Each COSEM logical device shall be identified by its Logical Device Name, unique worldwide. See 4.1.8.2. It is held by the *value* attribute of a "Data" object, with data type *octet-string* or *visible-string*. For short name referencing, the base_name of the object is fixed. See 4.1.3.

COSEM logical device name object	IC	OBIS code					
		A	B	C	D	E	F
COSEM logical device name	1, Data [a]	0	0	42	0	0	255
[a] If the IC "Data" is not available, "Register" (with scaler = 0, unit = 255) may be used.							

6.2.36 Information security related objects

Instances of the IC "Security setup" – see 4.4.7 – are used to set up the message security features. For each Association object, there is one Security setup object managing security within that AA. See 5.4.2 and 5.4.3. Value group E numbers the instances.

Security setup objects	IC	OBIS code					
		A	B	C	D	E	F
Security setup	64, Security setup	0	0	43	0	*e*	255

Invocation counter objects hold the invocation counter element of the initialization vector. They are instances of the IC "Data". The value in value group B identifies the communication channel.

NOTE The same client may use different communication channels, e.g. a remote port and a local port. The invocation counter on the different channels may be different.

The value in value group E shall be the same as in the logical name of the corresponding "Security setup" object.

Invocation counter objects	IC	OBIS code					
		A	B	C	D	E	F
Invocation counter	1, Data [a]	0	*b*	43	1	*e*	255
NOTE In earlier versions of this standard, these objects were called Frame counter objects.							
[a] If the IC "Data" is not available, "Register" (with scaler = 0, unit = 255) may be used.							

Instances of the IC "Data protection" – see 4.4.9 – are used to apply / remove protection on COSEM data, i.e. sets of attributes values, method invocation and return parameters. Value group E numbers the instances.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Data protection objects	IC	OBIS code					
		A	B	C	D	E	F
Data protection	30, Data protection	0	0	43	2	*e*	255

6.2.37 Image transfer objects (class_id = 18)

Instances of the IC “Image transfer” – see 4.4.6 – control the Image transfer process.

Image transfer related objects	IC	OBIS code					
		A	B	C	D	E	F
Image transfer	18, Image transfer	0	0	44	0	*e*	255

6.2.38 Function control objects (class_id = 122)

Instances of the IC “Function control“ – see 4.4.10 – allow enabling and disabling functions in the server.

Function control related objects	IC	OBIS code					
		A	B	C	D	E	F
Function control	122, Function control	0	0	44	1	*e*	255

6.2.39 Communication port protection objects (class_id = 124)

Instances of the “Communication port protection” IC – see 4.4.12 – control the communication port protection mechanism.

Communication port protection objects	IC	OBIS code					
		A	B	C	D	E	F
Communication port protection objects	124	0	*b*	44	2	*e*	255

6.2.40 Utility table objects (class_id = 26)

Instances of the IC “Utility tables” – see 4.3.7 – allow representing ANSI utility tables. The Utility table IDs are mapped to OBIS codes as follows:

- value group A: use value of 0 to specify abstract object;
- value group B: instance of table set;
- value group C: use value 65 – signifies utility tables specific definitions;
- value group D: table group selector;
- value group E: table number within group;
- value group F: use value 0xFF for data of current billing period.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Utility table objects	IC	OBIS code					
		A	B	C	D	E	F
Standard tables 0-127	26, Utility tables	0	*b*	65	0	*e*	255
Standard tables 128-255		0	*b*	65	1	*e*	255
…							
Standard tables 1920-2047		0	*b*	65	15	*e*	255
Manufacturer tables 0-127		0	*b*	65	16	*e*	255
Manufacturer tables 128-255		0	*b*	65	17	*e*	255
…							
Manufacturer tables 1920-2047		0	*b*	65	31	*e*	255
Std pending tables 0-127		0	*b*	65	32	*e*	255
Std pending tables 128-255		0	*b*	65	33	*e*	255
…							
Std pending tables 1920-2047		0	*b*	65	47	*e*	255
Mfg pending tables 0-127		0	*b*	65	48	*e*	255
Mfg pending tables 128-255		0	*b*	65	49	*e*	255
…							
Mfg pending tables 1920-2047		0	*b*	65	63	*e*	255

6.2.41 Compact data objects (class_id = 62)

"Compact data" objects – see 4.3.10 – store data and metadata separated, thus they allow reducing overhead.

Compact data objects	IC	OBIS code					
		A	B	C	D	E	F
Compact data	62, Compact data	0	*b*	66	0	e	255

6.2.42 Device ID objects

A series of objects are used to hold ID numbers of the device. These ID numbers can be defined by the manufacturer (e.g. manufacturing number) or by the user.

They are held by the *value* attribute of "Data" objects, with data type *double-long-unsigned*, *octet-string*, *visible-string, utf8-string*, *unsigned, long-unsigned.* If more than one of those is used, it is allowed to combine them into a "Profile generic" object. In this case, the captured objects are *value* attributes of the device ID "Data" objects, the capture period is 1 to have just actual values, the sort method is FIFO and the profile entries are limited to 1. Alternatively, a "Register table" object – see 4.3.8 – can be used. See also IEC 62056-6-1:2023, Table 8.

Device ID objects	IC	OBIS code					
		A	B	C	D	E	F
Device ID 1…10 object (manufacturing number)	1, Data [a]	0	*b*	96	1	0…9	255
Device ID-s object	7, Profile generic	0	*b*	96	1	255	255
Device ID-s object	61, Register table	0	*b*	96	1	255	255

[a] If the IC "Data" is not available, "Register" or "Extended register" (with scaler = 0, unit = 255) may be used.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.43 Metering point ID objects

One object is available to store a media type independent metering point ID. It is held by the *value* attribute of a “Data” object, with data type *double-long-unsigned*, *octet-string*, *visible-string, utf8-string*, *unsigned, long-unsigned.*

Metering point ID objects	IC	OBIS code					
		A	B	C	D	E	F
Metering point ID	1, Data [a]	0	*b*	96	1	10	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.2.44 Parameter changes and calibration objects

A set of simple COSEM objects describes the history of the configuration of the device. All values are modelled by instances of the IC "Data".

Parameter changes objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data [a]	0	*b*	96	2	*e*	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.2.45 I/O control signal objects

A series of objects are available to define and control the status of I/O lines of the physical metering equipment.

The status is held by the *value* attribute of a “Data” object, with data type *octet-string* or *bit-string.* Alternatively, the status is held by a “Status mapping” object, see 4.3.9, which holds both the status word and the mapping of its bits to the reference table. If there are several I/O control status objects used, it is allowed to combine them into an instance of the IC “Profile generic” or “Register table”, using the OBIS code of the global state of I/O control signals object. See also IEC 62056-6-1:2023, Table 8.

I/O control signal objects	IC	OBIS code					
		A	B	C	D	E	F
I/O control signal objects, contents manufacturer specific	1, Data [a]	0	*b*	96	3	0…4	255
I/O control signal objects, contents mapped to a reference table	63, Status mapping	0	*b*	96	3	0…4	255
I/O control signal objects, global	7, Profile generic or 61, Register table	0	*b*	96	3	0	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.2.46 Disconnect control objects (class_id = 70)

Instances of the IC “Disconnect control” – see 4.5.8 – manage internal or external disconnect units (e.g. electricity breaker, gas valve) in order to connect or disconnect – partly or entirely – the premises of the consumer to / from the supply. See also 6.2.22.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Disconnect control objects	IC	OBIS code					
		A	B	C	D	E	F
Disconnect control	70, Disconnect control	0	b	96	3	10	255

6.2.47 Arbitrator objects (class_id = 68)

Instances of the IC “Arbitrator” – see 4.5.12 – are used

Arbitrator Objects	IC	OBIS code					
		A	B	C	D	E	F
General-purpose Arbitrator	68, Arbitrator	0	*b*	96	3	*20 ... 29*	255

6.2.48 Status of internal control signals objects

A series of objects are available to hold the status of internal control signals.

The status carries binary information from a bitmap, and it shall be held by the *value* attribute of a “Data” object, with data type *bit-string, unsigned, long-unsigned, double-long-unsigned, long64-unsigned or octet-string.* Alternatively, the status is held by a “Status mapping” object, see 4.3.9, which holds both the status word and the mapping of its bits to the reference table. If there are several status of internal control signals objects used, it is allowed to combine them into an instance of the IC “Profile generic” or “Register table”, using the OBIS code of the global “Internal control signals” object. See also IEC 62056-6-1:2023, Table 8.

Internal control signals objects	IC	OBIS code					
		A	B	C	D	E	F
Internal control signals, contents manufacturer specific	1, Data [a]	0	*b*	96	4	0...4	255
Internal control signals, contents mapped to a reference table	63, Status mapping	0	*b*	96	4	0...4	255
Internal control signals, global	7, Profile generic or 61, Register table	0	*b*	96	4	0	255

[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.

6.2.49 Internal operating status objects

A series of objects are available to hold internal operating statuses.

The status carries binary information from a bitmap, and it shall be held by the value attribute of a “Data” object, with data type *bit-string, unsigned, long-unsigned, double-long-unsigned, long64-unsigned or octet-string.* Alternatively, the status is held by a “Status mapping” object, see 4.3.9, which holds both the status word and the mapping of its bits to the reference table. If there are several status of internal control signals objects used, it is allowed to combine them into an instance of the IC “Profile generic” or “Register table”, using the OBIS code of the global ”Internal operating status” object. See also IEC 62056-6-1:2023, Table 8.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Internal operating status objects	IC	OBIS code					
		A	B	C	D	E	F
Internal operating status objects, contents manufacturer specific	1, Data [a]	0	*b*	96	5	0…4	255
Internal operating status objects, contents mapped to a reference table	63, Status mapping	0	*b*	96	5	0…4	255
Internal operating status objects, global	7, Profile generic or 61, Register table	0	*b*	96	5	0	255
[a] If the IC "Data" is not available, "Register" or "Extended register" (with scaler = 0, unit = 255) may be used.							

Internal operating status objects can also be related to an energy type. See 6.3.7.

6.2.50 Battery entries objects

A series of objects are available for holding information relative to the battery of the device. These objects are instances of IC "Data", "Register" or "Extended register" as appropriate.

Battery entries objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register or 4, Extended register	0	*b*	96	6	0…6	255

6.2.51 Power failure monitoring objects

A series of objects are available for power failure monitoring:

- For simple power failure monitoring, it is possible to count the number of power failure events affecting all three phases, one of the three phases, any of the phases, and the auxiliary supply;
- For advanced power failure monitoring, it is possible to define a time threshold to make a distinction between short and long power failure events. It is possible to count the number of such long power failure events separately from the short ones, as well as to store their time of occurrence and duration (time from power down to power up) in all three phases, in one of the three phases and in any of the phases;
- The number of power failure events objects are represented by instances of the IC "Data", "Register" or "Extended register" with data types *unsigned*, *long-unsigned*, *double-long-unsigned* or *long64-unsigned*;
- The power failure duration, time and time threshold data are represented by instances of the IC "Data", "Register" or "Extended register" with appropriate data types;
- If power failure duration objects are represented by instances of the IC "Data", then the default scaler shall be 0, and the default unit shall be the second.

Power failure monitoring objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register or 4, Extended register	0	*b*	96	7	0… 21	255

These objects may be collected in a "Power failure event log" object. See IEC 62056-6-1:2023, Table 23.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.52 Operating time objects

A series of objects are available for holding the cumulated operating time and the various tariff registers of the device. These objects are instances of the IC “Data”, ”Register” or “Extended register”. The data type shall be *unsigned*, *long-unsigned* or *double-long-unsigned* with appropriate scaler and unit. If the IC “Data” is used, the unit shall be the second by default.

Operating time objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023,Table 8.	1, Data, 3, Register or 4, Extended register	0	*b*	96	8	0... 63	255

6.2.53 Environment related parameters objects

A series of objects are available to store environmental related parameters. They are held by the *value* attribute of instances of the IC “Register” or “Extended register”, with appropriate data types.

Environment related parameters objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023,Table 8.	3, Register or 4, Extended register	0	*b*	96	9	0...2	255

6.2.54 Status register objects

A series of objects are available to hold statuses that can be captured in load profiles. See also see IEC 62056-6-1:2023, Table 8.

Status register objects	IC	OBIS code					
		A	B	C	D	E	F
Status register, contents manufacturer specific	1, Data [a]	0	*b*	96	10	1... 10	255
Status register, contents mapped to reference table	63, Status mapping	0	*b*	96	10	1... 10	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

The status register is held by the value attribute of a “Data” object, with data type *bit-string, unsigned, long-unsigned, double-long-unsigned, long64-unsigned or octet-string.* It carries binary information from a bitmap. It’s contents is not specified.

Alternatively, the status register may be held by the *status_word* attribute of a “Status mapping” object, see 4.3.9. The *mapping_table* attribute holds mapping information between the bits of the status word and entries of a reference table.

6.2.55 Event code objects

In the meter or in its environment, various events may be generated. A series of objects are available to hold an identifier of a most recent event (event code). Different instances of event code objects may be captured in different instances of event logs; see 6.2.66.

NOTE The definition of event identifiers is out of the Scope of this document.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Event code objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	11	0… 99	255

Events may also set flags in error registers and alarm registers. See also 6.2.64.

6.2.56 Communication port log parameter objects

A series of objects are available to hold various communication log parameters. They are represented by instances of IC “Data”, “Register” or “Extended register”.

Communication port log parameter objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	12	0…6	255

6.2.57 Consumer message objects

A series of objects are available to store information sent to the energy end-user. The information may appear on the display of the meter and / or on a consumer information port.

Consumer message objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	13	0, 1	255

6.2.58 Currently active tariff objects

A series of objects are available to hold the identifier of the currently active tariff. They carry the same information as the *active_mask* attribute of the corresponding “Register activation” object.

Currently active tariff objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	14	0… 15	255

6.2.59 Event counter objects

A series of objects are available to count events. The number of the events is held by the value attribute.

Event counter objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	15	0… 99	255

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.60 Profile entry digital signature objects

Instances of “Data”, “Register” or “Extended register” objects hold digital signatures of “Profile generic” object buffer entries. If the *capture_object* attribute of a “Profile generic” object contains a reference to a “Profile entry digital signature” object, then the digital signature is calculated and captured together with the other attribute values.

The security context is determined by the “Security setup” object which is which is visible in the same AA (object_list).

NOTE The digital signature may be generated when the entry is captured or “on the fly”, when an entry is accessed.

Profile entry digital signature objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	16	0…9	255

6.2.61 Profile entry counter objects

A series of objects are available to keep track of the total number of entries captured in a “Profile generic” object. These objects are instances of the IC “Data”, Register” or “Extended register”. The data type shall be *unsigned*, *long-unsigned*, *double-long-unsigned or long64-unsigned* with appropriate scaler and unit. If the *capture_objects* attribute of a “Profile generic” object contains a reference to a “Profile entry counter” object, then the counter is monotonically increased by one and captured together with the other attribute values when the *capture* method is invoked or when auto capture takes place.

If a “Profile entry counter” object reaches its maximum value, it will wrap-around and the count will re-start at zero.

Profile entry counter objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	17	0… 127	255

6.2.62 Meter tamper event related objects

A series of objects are available to register characteristics of various meter tamper events. These objects are instances of the IC “Data”, Register” or “Extended register”. The data type shall be *unsigned*, *long-unsigned or double-long-unsigned* with appropriate scaler and unit. For time stamps, the data type shall be *double-long-unsigned* (in the case of UNIX time), *octet-string* or *date-time* formatted as specified in 4.1.6.1.

- Meter open events are related to cases when the meter case is open;
- Terminal cover open events are related to cases when a terminal cover is removed (open);
- Tilt events are related to cases when the meter is not in its normal operation position;
- Strong DC magnetic field events are related to cases when the presence of a strong DC magnetic field is detected;
- Metrology tamper events are related to cases when an anomaly in the operation of the metrology is detected due to a perceived tamper;
- Communication tamper events are related to cases when an anomaly in the operation of the communication interfaces is detected due to a perceived tamper.

The method of detecting the various tampers is out of the Scope of this document.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Meter tamper event related objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 8.	1, Data, 3, Register, or 4, Extended register	0	*b*	96	20	*e*	255

6.2.63 Error register objects

A series of objects are used to communicate error indications of the device. The different error registers are held by the *value* attribute of “Data” objects, with data type or *bit-string*, *octet-string, unsigned, long-unsigned, double-long-unsigned* or *long64-unsigned.*

The individual bits of the error register may be set and cleared by a pre-defined selection of events – see 6.2.55. Depending on the type of the error, some errors may clear themselves when the reason setting the error flag disappears.

If more than one of those objects is used, it is allowed to combine them into one instance of the IC "Profile generic". In this case, the captured objects are the *value* attributes “Data” objects, the capture period is 1 to have just actual values, the sort method is FIFO, the profile entries are limited to 1. Alternatively, an instance of the IC “Register table” can be used.

Error register objects can also be related to an energy type and to a channel. See IEC 62056-6-1:2023, 6.2 and 7.5.2.

Error register objects	IC	OBIS code					
		A	B	C	D	E	F
Error register 1…10 object	1, Data [a]	0	b	97	97	0…9	255
Error profile object	7, Profile generic	0	b	97	97	255	255
Error table object	61, Register table	0	b	97	97	255	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.2.64 Alarm register, Alarm filter and Alarm descriptor objects

A number of objects are available to hold alarm registers. The different alarm registers are held by the value attribute of “Data” objects, with data type *bit-string*, *octet-string*, *unsigned*, *long-unsigned*, *double-long-unsigned* or *long64-unsigned*. When selected events occur, they set the corresponding flag and the device may raise an alarm. Depending on the type of alarm, some alarms may clear themselves when the reason setting the alarm flag disappears.

If more than one of those objects is used, it is also allowed to combine them into one instance of the IC "Profile generic". In this case, the captured objects are the *value* attributes of “Data” objects, the capture period is 1 to have just actual values, the sort method is FIFO, and the profile entries are limited to 1. Alternatively, an instance of the IC “Register table” can be used.

Alarm filter objects are available to define if an event is to be handled as an alarm when it appears. The different alarm filters are held by the value attribute of “Data” objects, with data type *bit-string, octet-string, unsigned, long-unsigned, double-long-unsigned* or *long64-unsigned*. The bit mask has the same structure as the corresponding alarm register object. If a bit in the alarm filter is set, then the corresponding alarm is enabled, otherwise it is disabled. *Alarm filter* objects act on *Alarm register* and *Alarm descriptor* objects the same way.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Alarm descriptor objects are available to persistently hold the occurrence of alarms. The different alarm descriptors are of the same type as the corresponding *Alarm register*. When a selected event occurs, the corresponding flag is set in the *Alarm register* as well as in the *Alarm descriptor* objects. An alarm descriptor flag remains set even if the corresponding alarm condition has disappeared. Alarm descriptor flags do not reset themselves; they can be reset by writing the value attribute only.

NOTE The alarm conditions, the structure of the *Alarm register / Alarm filter / Alarm descriptor* objects are subject to a project specific companion specification.

Alarm register, Alarm filter and Alarm descriptor objects	IC	OBIS code					
		A	B	C	D	E	F
Alarm register objects 1…10	1, Data [a]	0	b	97	98	0…9	255
Alarm register profile object	7, Profile generic	0	b	97	98	255	255
Alarm register table object	61, Register table	0	b	97	98	255	255
Alarm filter objects 1…10	1, Data [a]	0	b	97	98	10…19	255
Alarm descriptor objects 1…10	1, Data [a]	0	b	97	98	20…29	255
[a] If the IC “Dtaata” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.2.65 General list objects

Instances of the IC “Profile generic” are used to model lists of any kind of data, for example measurement values, constants, statuses, events. They are modelled by “Profile generic” objects. One standard object per billing period scheme is defined.

List objects may be also related to an energy type and to a channel.

General list objects	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, 6.3 and 7.5.3	7, Profile generic	0	*b*	98	*d*	*e*	255[a]
[a] F = 255 means a wildcard here. See IEC 62056-6-1:2023,Clause A.3.							

6.2.66 Event log objects (class_id 7)

Instances of the IC “Profile generic” are used to store Event logs. Event logs may be also media related. In this case, the value of value group A shall be the relevant media identifier. See also IEC 62056-6-1:2023, 6.5 and 7.5.4.

Event log objects	IC	OBIS code					
		A	B	C	D	E	F
Event log	7, Profile generic	a	b	99	98	e	255[a]
[a] F = 255 means a wildcard here. See IEC 62056-6-1:2023, Clause A.3.							
NOTE 1 Event logs may capture for example the time of occurrence of the event, the event code and other relevant data. NOTE 2 Project specific companion specifications may specify a more precise meaning of the instances of the different event logs, i.e. the data captured and the number of events captured.							

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.67 Inactive objects

Inactive objects are objects, which are present in the meter, but which do not have an assigned functionality. Inactive instances of any IC may be present. See also IEC 62056-6-1:2023, 5.3.2.

Inactive objects	IC	OBIS code					
		A	B	C	D	E	F
Inactive objects	Any	0	*b*	127	0	*e*	255

6.3 Electricity related COSEM objects

6.3.1 Value group D definitions

The different ways of processing measurement values as defined by value group D – see IEC 62056-6-1:2023, 7.2.1 – are modelled as shown in Table 53.

Table 53 – Representation of various values by appropriate ICs

Type of value	Represented by
cumulative values	Instances of IC "Register" or “Extended register”.
maximum and minimum values	Instances of IC "Profile generic" with sorting method *maximum* or *minimum*, depth according to implementation and captured objects according to implementation. A single maximum value or minimum value can alternatively be represented by an instance of the IC “Register” or "Extended register”.
current and last average values	Instances of IC “Demand register”. The logical name is the OBIS code of the current average value (D = 4, 14, or 24). For display purposes: Instances of IC “Register” or “Demand register”. The logical name is the OBIS code of current average (D = 4, 14 or 24) or last average (D = 5, 15 or 25) as appropriate.
instantaneous values	Instances of IC "Register”.
time integral values	Instances of IC "Register” or “Extended register”.
occurrence counters	Instances of IC “Data” or "Register".
contracted values	Instances of IC "Register" or “Extended register”.
Under/Over limit thresholds	Instances of IC "Register" or “Extended register”.
Over/Under limit occurrence counters	Instances of IC "Register" or “Extended register”.
Under/Over limit durations	Instances of IC "Register" or “Extended register”.
Over/Under limit magnitudes	Instances of IC "Register" or “Extended register”.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.3.2 ID numbers – Electricity

The different electricity ID numbers are held by instances of the IC "Data”, with data type *unsigned, long-unsigned, double-long-unsigned, octet-string* or *visible-string.* If more than one of those is used, it is allowed to combine them into a "Profile generic" object. In this case, the captured objects are *value* attributes of electricity ID “Data” objects, the capture period is 1 to have just actual values, the sort method is FIFO, the profile entries are limited to 1. Alternatively, a “Register table” object can be used. See also IEC 62056-6-1:2023, Table 20.

Electricity ID objects	IC	OBIS code					
		A	B	C	D	E	F
Electricity ID 1…10 object	1, Data [a]	1	*b*	0	0	0…9	255
Electricity ID-s object	7, Profile generic	1	*b*	0	0	255	255
Electricity ID-s object	61, Register table	1	*b*	0	0	255	255
[a] If the IC "Data" is not available, "Register" or "Extended register" (with scaler = 0, unit = 255) may be used.							

6.3.3 Billing period values / reset counter entries

These values are represented by instances of the IC "Data".

For billing period / reset counters and for number of available billing periods the data type shall be *unsigned*, *long-unsigned* or *double-long-unsigned*. For time stamps of billing periods, the data type shall be *double-long-unsigned* (in the case of UNIX time), *octet-string* or *date-time* formatted as specified in 4.1.6.1.

These objects may be related to channels.

When the values of historical periods are represented by "Profile generic" objects, the time stamp of the billing period objects shall be part of the captured objects.

Billing period values / reset counter entries	IC	OBIS code					
		A	B	C	D	E	F
For item names and OBIS codes see IEC 62056-6-1:2023, Table 20.	1, Data [a]	1	*b*	0	1	*e*	255
[a] If the IC "Data" is not available, "Register" or "Extended register" (with scaler = 0, unit = 255) may be used.							

6.3.4 Other electricity related general purpose objects

Program entries shall be represented by instances of the IC "Data" with data type *unsigned, long-unsigned, octet-string* or *visible-string.* For "Meter connection diagram ID" objects data type *enumerated* can be used as well. Program entries can also be related to a channel.

Output pulse constant, reading factor, CT/VT ratio, nominal value, input pulse constant, transformer and line loss coefficient values shall be represented by instances of the IC "Data", "Register" or "Extended register". For the *value* attribute, only simple data types are allowed.

Measurement period, recording interval and billing period duration values shall be represented by instances of IC "Data", "Register" or "Extended register" with the data type of the *value* attribute *unsigned, long-unsigned* or *double-long-unsigned.* The default unit is the second.

Time entry values shall be represented by instances of IC "Data", "Register" or "Extended register" with the data type of the *value* attribute *octet-string*, formatted as *date-time* in 4.1.6.1. The data types *unsigned, integer, long-unsigned* or *double-long-unsigned* can also be used where appropriate.

The *Clock synchronization method* shall be represented by an instance of an IC "Data" with data type *enum*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Synchronization method	enum:	(0)	no synchronization,
		(1)	adjust to quarter,
		(2)	adjust to measuring period,
		(3)	adjust to minute,
		(4)	reserved,
		(5)	adjust to preset time,
		(6)	shift time

For the detailed OBIS codes, see IEC 62056-6-1:2023, Table 20.

Electricity related general purpose objects	**IC**	**OBIS code**					
		A	**B**	**C**	**D**	**E**	**F**
Program entries	1, Data [a]	1	*b*	0	2	*e*	255
Output pulse values or constants	1, Data	1	*b*	0	3	*e*	255
Reading factor and CT/VT ratio	3, Register 4, Extended Register	1	*b*	0	4	*e*	255
Nominal values	3, Register 4, Extended Register	1	*b*	0	6	*e*	255
Input pulse values or constants	1, Data 3, Register 4, Extended Register	1	*b*	0	7	*e*	255
Measurement period- / recording interval- / billing period duration	1, Data 3, Register 4, Extended Register	1	*b*	0	8	*e*	255
Time entries		1	*b*	0	9	*e*	255
Transformer and line loss coefficients	3, Register 4, Extended Register	1	*b*	0	10	*e*	255
[a] If the IC “Data” is not available, “Register” or “Extended register” (with scaler = 0, unit = 255) may be used.							

6.3.5 Measurement algorithm

These values are represented by instances of the IC “Data”, with data type *enum*.

Measurement algorithm objects	**IC**	**OBIS code**					
		A	**B**	**C**	**D**	E	F
Measuring algorithm for active power	1, Data [a]	1	*b*	0	11	1	255
Measurement algorithm for active energy		1	*b*	0	11	2	255
Measurement algorithm for reactive power		1	*b*	0	11	3	255
Measurement algorithm for reactive energy		1	*b*	0	11	4	255
Measurement algorithm for apparent power		1	*b*	0	11	5	255
Measurement algorithm for apparent energy		1	*b*	0	11	6	255
Measurement algorithm for power factor calculation		1	*b*	0	11	7	255
[a] In cases where the IC “Data” is not available, “Register” (with scaler = 0, unit = 255) may be used.							

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The enumerated values are specified in Table 54:

Table 54 – Measuring algorithms – enumerated values

Measuring algorithm for active power and energy	
(0)	not specified
(1)	only the fundamentals of voltage and current are used
(2)	all harmonics of voltage and current are used
(3)	only the DC part of voltage and current is used
(4)	all harmonics and the DC part of voltage and current are used
Measuring algorithm for reactive power and energy	
(0)	not specified
(1)	(sum of) reactive power of each phase, calculated from the fundamental of the per phase voltage and the per phase current
(2)	polyphase reactive power calculated from polyphase apparent power and polyphase active power
(3)	(sum of) reactive power calculated from per phase apparent power and per phase active power
Measurement algorithm for apparent power and energy	
(0)	not specified
(1)	$S = U \times I$, with voltage: only fundamental, and current: only fundamental
(2)	$S = U \times I$, with voltage: only fundamental, and current: all harmonics
(3)	$S = U \times I$, with voltage: only fundamental, and current: all harmonics and DC part
(4)	$S = U \times I$, with voltage: all harmonics, and current: only fundamental
(5)	$S = U \times I$, with voltage: all harmonics, and current: all harmonics
(6)	$S = U \times I$, with voltage: all harmonics, and current: all harmonics and DC part
(7)	$S = U \times I$, with voltage: all harmonics and DC part, and current: only fundamental
(8)	$S = U \times I$, with voltage: all harmonics and DC part, and current: all harmonics
(9)	$S = U \times I$, with voltage: all harmonics and DC part, and current: all harmonics and DC part
(10)	$S = \sqrt{P^2 + Q^2}$, with P: only fundamental in U and I, and Q: only fundamental in U and I, where P and Q are polyphase quantities
(11)	$S = \sqrt{P^2 + Q^2}$, with P: all harmonics in U and I, and Q: only fundamental in U and I where P and Q are polyphase quantities
(12)	$S = \sqrt{P^2 + Q^2}$, with P: all harmonics and DC part in U and I, and Q: only fundamental in U and I where P and Q are polyphase quantities
(13)	$S = \Sigma\sqrt{P^2 + Q^2}$, with P: only fundamental in U and I, and Q: only fundamental in U and I where P and Q are single phase quantities
(14)	$S = \Sigma\sqrt{P^2 + Q^2}$, with P: all harmonics in U and I, and Q: only fundamental in U and I where P and Q are single phase quantities
(15)	$S = \Sigma\sqrt{P^2 + Q^2}$, with P: all harmonics and DC part in U and I, and Q: only fundamental in U and I where P and Q are single-phase quantities
Measurement algorithm for power factor calculation	
(0)	not specified
(1)	displacement power factor: the displacement between fundamental voltage and current vectors, which can be calculated directly from fundamental active power and apparent power, or another appropriate algorithm,
(2)	true power factor, the power factor produced by the voltage and current, including their harmonics . It may be calculated from apparent power and active power, including the harmonics.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.3.6 Metering point ID (electricity related)

A series of objects are available to hold electricity related metering point IDs. They are held by the *value* attribute of "Data" objects, with data type *unsigned, long-unsigned, double-long unsigned*, *octet-string* or *visible-string.* If more than one of those is used, it is allowed to combine them into one instance of the IC "Profile generic". In this case, the captured objects are the *value* attributes of the electricity related metering point ID "Data" objects, the capture period is 1 to have just actual values, the sort method is FIFO, the profile entries are limited to 1. Alternatively, an instance of the IC "Register table" can be used. For detailed OBIS codes, see IEC 62056-6-1:2023, Table 20.

Metering point ID objects	IC	OBIS code					
		A	B	C	D	E	F
Metering point ID 1…10 (electricity related)	1, Data [a]	1	*b*	96	1	0…9	255
Metering point ID-s object	7, Profile generic	1	*b*	96		255	255
Metering point ID-s object	61, Register table	1	*b*	96	1	255	255
[a] If the IC "Data" is not available, "Register" (with scaler = 0, unit = 255) may be used.							

6.3.7 Electricity related status objects

A number of electricity related objects are available to hold information about the internal operating status, the starting of the meter and the status of voltage and current circuits.

The status is held by the value attribute of a "Data" object, with data type *bit-string, unsigned, long-unsigned, double-long-unsigned, long64-unsigned or octet-string.*

Alternatively, the status is held by a "Status mapping" object, which holds both the status word and the mapping of its bits to the reference table.

If there are several electricity related internal operating status objects used, it is allowed to combine them into an instance of the IC "Profile generic" or "Register table", using the OBIS code of the global internal operating status. For detailed OBIS codes, see IEC 62056-6-1:2023, Table 20.

Electricity related status objects	IC	OBIS code					
		A	B	C	D	E	F
Internal operating status signals, electricity related, contents manufacturer specific	1, Data [a]	1	*b*	96	5	0…5	255
Internal operating status signals, electricity related, contents mapped to reference table	63, Status mapping	1	*b*	96	5	0…5	255
Electricity related status data, contents manufacturer specific	1, Data [a]	1	*b*	96	10	0…3	255
Electricity related status data, contents mapped to reference table	63, Status mapping	1	*b*	96	10	0…3	255
[a] If the IC "Data" is not available, "Register" or "Extended register" (with scaler = 0, unit = 255) may be used.							

6.3.8 List objects – Electricity (class_id = 7)

These COSEM objects are used to model lists of any kind of data, for example measurement values, constants, statuses, events. They are modelled by "Profile generic" objects.

One standard object per billing period scheme is defined. See also IEC 62056-6-1:2023, 7.5.3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

List objects – Electricity	IC	OBIS code					
		A	B	C	D	E	F
For names and OBIS codes see IEC 62056-6-1:2023, Table 22.	7, Profile generic	1	*b*	98	*d*	*e*	255[a]
[a] F = 255 means a wildcard here. See IEC 62056-6-1:2023, Clause A.3.							

6.3.9 Threshold values

A number of objects are available for representing thresholds for instantaneous quantities. The thresholds may be “under limit”, “over limit”, “missing” and “time thresholds”. Time thresholds are used to detect “under limit”, “over limit” and “missing” conditions.

Objects are also available to represent the number of occurrences when these thresholds are exceeded, the duration of such events and the magnitude of the quantity during such events.

These values are represented by instances of IC “Data”, “Register” or “Extended register”.

All these quantities may be related to tariffs.

As defined in IEC 62056-6-1:2023, 7.4.2, value group F may be used to identify multiple thresholds.

For OBIS codes, see Table 55 and IEC 62056-6-1:2023, Table 14.

Table 55 – Threshold objects, electricity

Threshold objects	IC	OBIS code					
		A	B	C	D	E	F
Threshold objects for instantaneous values	1, Data, 3, Register, 4, Extended register	1	*b*	1…10, 13, 14, 16…20, 21…30, 33, 34, 36…40, 41…50, 53, 54, 56…60, 61…70, 73, 74, 76…80, 82, 84…89	31…34, 35…38, 39…42, 43…45	0…63	0…99. 255
Threshold objects for harmonics of voltage, current and active power		1	*b*	11, 12, 15, 31, 32, 35, 51, 52, 55, 71, 72, 75, 90…92		0…120, 124… 127	

For monitoring the supply voltage, a more sophisticated functionality is also available, that allows counting the number of occurrences classified by the duration of the event and the depth of the voltage dip. For OBIS codes, see IEC 62056-6-1:2023, Table 8.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.3.10 Register monitor objects (class_id = 21)

Further to 6.2.13, the following definitions apply:

- For monitoring thresholds of instantaneous values, the logical name of the “Register monitor” object may be the OBIS identifier of the threshold;
- For monitoring current average and last average values, the logical name of the “Register monitor” object may be the OBIS identifier of the demand value monitored.

See Table 56.

Table 56 – Register monitor objects, electricity

<table>
<tr><th rowspan="2">Register monitor objects</th><th rowspan="2">IC</th><th colspan="6">OBIS code</th></tr>
<tr><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td rowspan="2">Instantaneous values, under limit / over limit / missing</td><td rowspan="4">21, Register monitor</td><td>1</td><td>b</td><td>c1</td><td rowspan="2">31, 35, 39</td><td>0-63</td><td rowspan="4">0-99, 255</td></tr>
<tr><td>1</td><td>b</td><td>c2</td><td>0-120, 124-127</td></tr>
<tr><td rowspan="2">Current average and last average values</td><td>1</td><td>b</td><td>c1</td><td rowspan="2">4, 5, 14, 15, 24, 25</td><td>0-63</td></tr>
<tr><td>1</td><td>b</td><td>c2</td><td>0-120, 124-127</td></tr>
<tr><td colspan="8">c1 = 1-10, 13,14, 16-20, 21-30, 33,34, 36-40, 41-50, 53,54, 56-60, 61-70, 73,74, 76-80, 82, 84-89.
c2 = 11, 12, 15, 31, 32, 35, 51, 52, 55, 71, 72, 75, 90-92, 100-107.</td></tr>
</table>

For the use of value group D, see IEC 62056-6-1:2023, Table 14.

For the use of value group E, see IEC 62056-6-1:2023, Table 15 and Table 16.

For the use of value group F, see IEC 62056-6-1:2023, 7.4.2.

6.4 Coding of OBIS identifications

To identify different instances of the same IC, their logical_name shall be different. In COSEM, the logical_name is taken from the OBIS definition (see 6.2, 6.3 and IEC 62056-6-1:2023).

OBIS codes are used within the COSEM environment as an *octet-string [6].* Each octet contains the unsigned value of the corresponding OBIS value group, coded without tags.

If a data item is identified by less than six value groups, all unused value groups shall be filled with 255.

Octet 1 contains the binary coded value of A (A = 0, 1, 2 ...9) in the four rightmost bits. The four leftmost bits contain the information on the identification system. The four leftmost bits set to zero indicate that the OBIS identification system (version 1) is used as *logical_name*.

Identification system used	Four leftmost bits of octet 1 (MSB left)
OBIS; see IEC 62056-6-1:2023.	0 0 0 0
Reserved for future use	0 0 0 1 … 1 1 1 1

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Within all value groups, the usage of a certain selection is fully defined; others are reserved for future use. If in the value groups B to F a value belonging to the manufacturer specific range (see IEC 62056-6-1:2023, 4.2) is used, then the whole OBIS code shall be considered as manufacturer specific, and the value of the other groups does not necessarily carry a meaning defined neither by Clause 5 of this standard nor by IEC 62056-6-1:2023.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Annex A
(informative)

Additional information on Auto answer and Auto connect ICs

NOTE This information is related to the “Auto answer” (class_id = 28, version = 2, see 4.7.5) and “Auto connect” (class_id = 29, version = 2, see 4.7.6) interface classes.

Since the capabilities (e.g. connection time, number of parallel connections) of communication networks (e.g. GPRS) are limited, devices e.g. meters are not permanently connected to the communication network.

Devices may connect to the network in regular intervals or on special events either to send unsolicited data or just to become accessible.

If a DLMS®/COSEM client e.g. a Head End System needs to access a server e.g. a meter that is not connected to the communication network a wake-up request can be sent. This may be a wake-up call or a wake-up message, e.g. an SMS message. After successfully processing the wake-up request the device connects to network.

Figure A.1 shows an example for a GSM/GPRS communication network. Please note that the dashed lines represent the network services, the solid lines refer to possible application layer services.

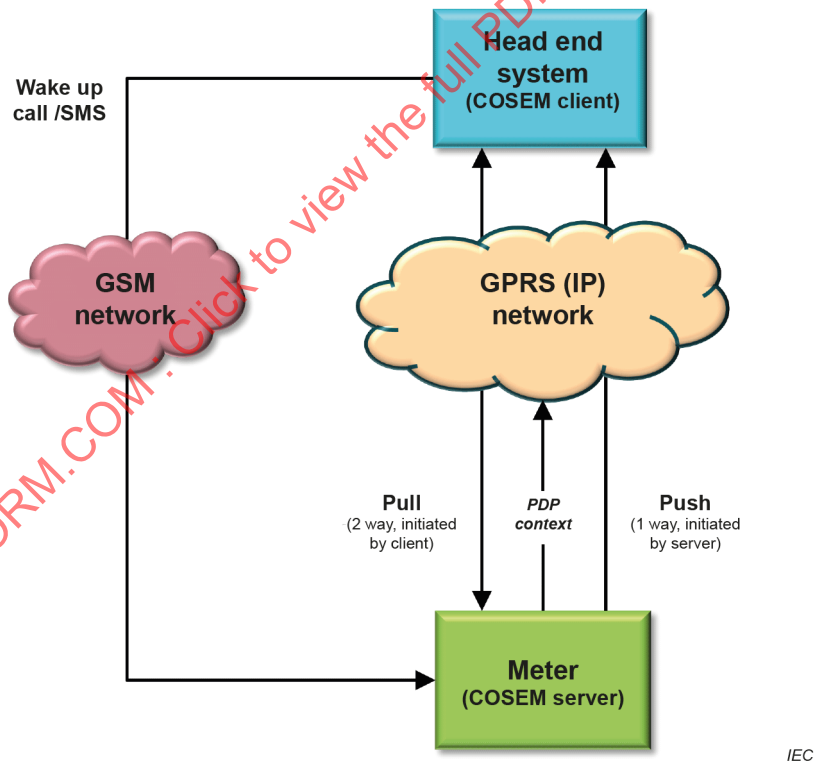


Figure A.1 – Network connectivity example for a GSM/GPRS network

The basic network connectivity in the case of a mobile network (GPRS or equivalent service) is modelled by the “Auto connect” IC. Depending on the mode the connection can be 'always on', 'always on in a time window' or 'only on after a wake-up'. If the device is connected to the network it has the PDP context attached and it is accessible from by the HES via its IP address. If necessary the current IP address of the server (meter) can be sent to the client (HES) using the DataNotification xDLMS service.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

The wake-up process is modelled using an instance of the “Auto answer” IC which provides additional security (check calling number) compared to today's solution.

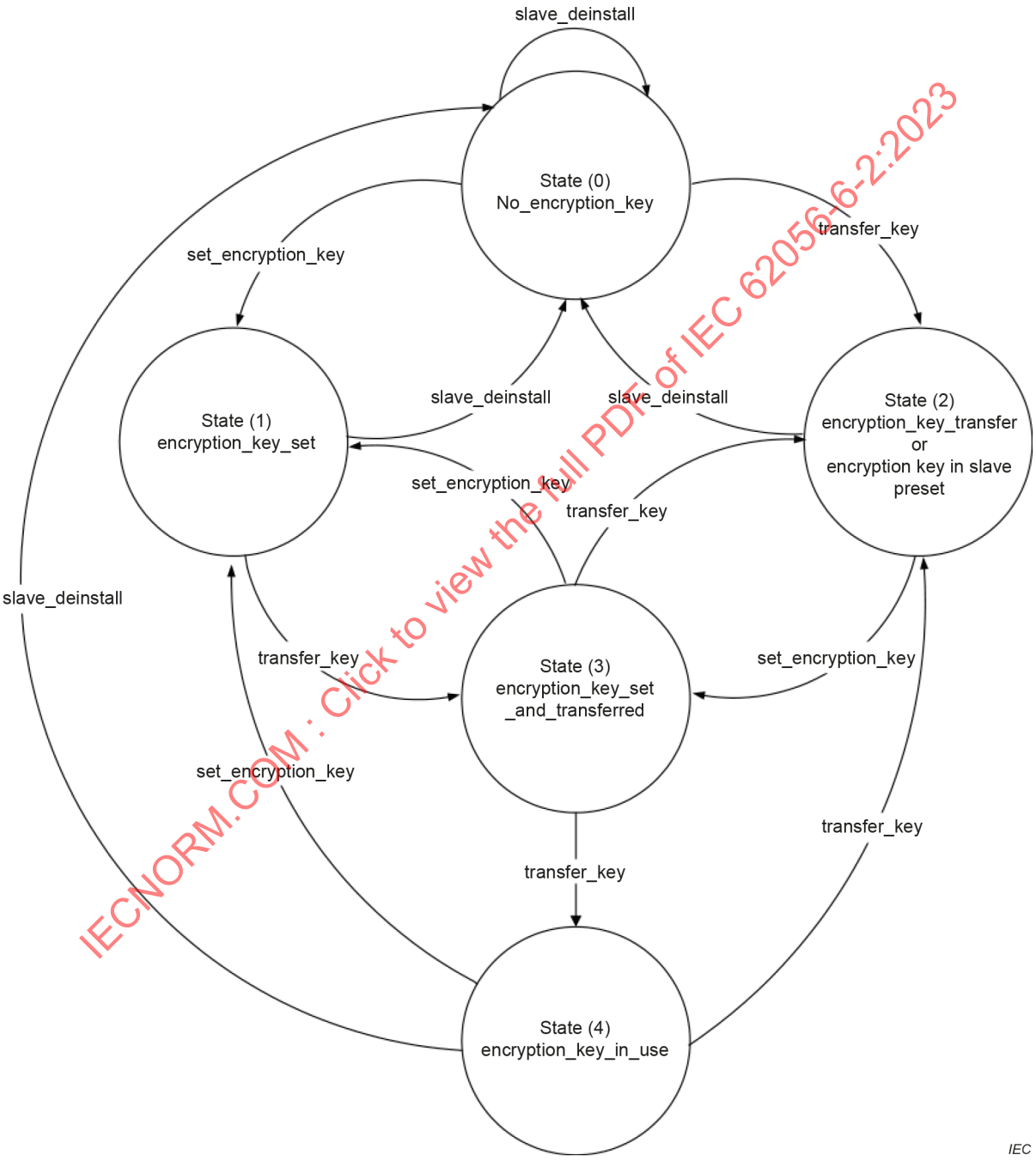
Also note that the “Auto answer” class is fully decoupled from the xDLMS application layer services. The main reason is to have a clear separation between the communication layers as well as to avoid creating an unsecured backdoor to execute application layer services with almost no protection. The execution of xDLMS services via SMS should be handled by sending ciphered xDLMS APDUs in a pre-established AA.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Annex B
(informative)

Additional information to M-Bus client (class_id = 72, version 1)

State transitions of the *encryption_key_status* attribute for different use cases are shown in Figure B.1.



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure B.1 – Encryption key status diagram

State transitions of the *encryption_key_status* attribute for different use cases are given below. At the time of installation of the slave four cases are possible:

a) Encryption key is preset in the slave and cannot be changed, see Table B.1;

b) Encryption key is preset in the slave and new key is set after installation, see Table B.2;

c) Encryption key is not preset in the slave, but can be set, see Table B.3 and Table B.4;

d) The slave is used without encryption. In this case, the encryption_key_status stays in state (0).

Table B.1 – Encryption key is preset in the slave and cannot be changed

Step	State	Condition for state transition (event or successfully invoked method)
1	(2)	set_encryption_key
2	(3)	encrypted telegrams successfully exchanged
3	(4)	–

Table B.2 – Encryption key is preset in the slave and new key is set after installation

Step	State	Condition for state transition (event or successfully invoked method)
1	(2)	set_encryption_key
2	(3)	transfer_key
3	(2)	set_encryption_key
4	(3)	encrypted telegrams successfully exchanged
5	(4)	–

Table B.3 – Encryption key is not preset in the slave, but can be set, case a)

Step	State	Condition for state transition (event or successfully invoked method)
1a	(0)	set_encryption_key
2a	(1)	transfer_key
3a	(3)	encrypted telegrams successfully exchanged
4a	(4)	–

Table B.4 – Encryption key is not preset in the slave, but can be set, case b)

Step	State	Condition for state transition (event or successfully invoked method)
1b	(0)	transfer_key
2b	(2)	set_encryption_key
3b	(3)	encrypted telegrams successfully exchanged
4b	(4)	–

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Annex C
(informative)

Additional information on IPv6 setup class (class_id = 48, version = 0)

C.1 General

In most regards, IPv6 is a conservative extension of IPv4. Most transport and application-layer protocols need little or no change to operate over IPv6; exceptions are application protocols that embed internet-layer addresses, such as FTP or NTPv3.

IPv6 specifies a new packet format, designed to minimize packet-header processing. Since the headers of IPv4 packets and IPv6 packets are significantly different, the two protocols are not interoperable.

C.2 IPv6 addressing

The most important feature of IPv6 is a much larger address space than that of IPv4: addresses in IPv6 are 128 bits long, compared to 32-bit addresses in IPv4. Furthermore, compared to IPv4, IPv6 supports multi-addressing on one physical interface (global, unique or link local IPv6 addresses).

IPv6 addresses are typically composed of two logical parts: a 64-bit (sub-)network prefix used for routing, and a 64-bit host part used to identify a host within the network.

The formats allowed for an IPv6 address are shown in Figure C.1.

(see http://www.iana.org/assignments/ipv6-address-space/). Note that to facilitate the IPv6 address writing, a specific notation defined in RFC 4291 has been specified by IETF (e.g. FF00::/8).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

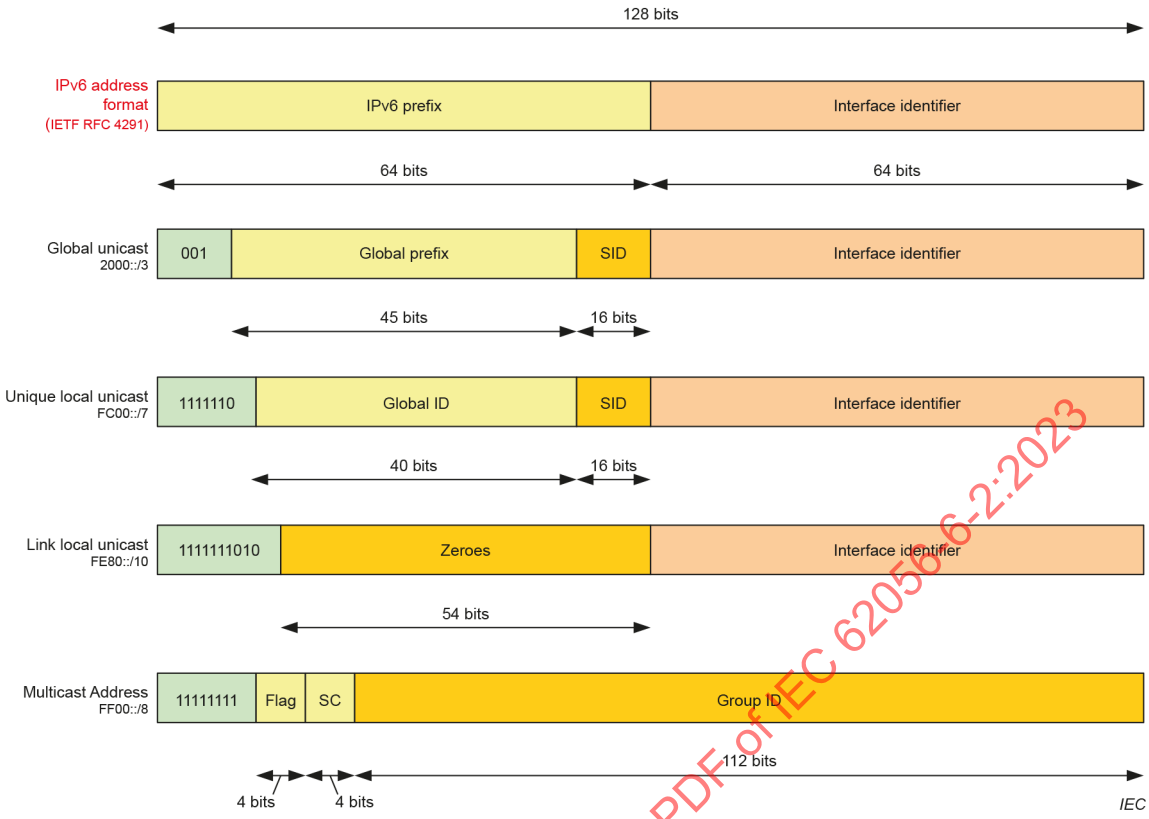


Figure C.1 – IPv6 address formats

Where:

– *Global Unicast* is a routable address in the whole internet network and is composed as follows:

  - Global prefix assigned by IANA (see http://www.iana.org/assignments/ipv6-unicast-address-assignments/);
  - Subnet ID (SID) allocated by the network administrator; and
  - Interface Identifier either generated from the interface's MAC address (using modified EUI-64 format), or obtained from a DHCPv6 server, or assigned manually;

– *Unique Local Unicast* is an address only applicable to local network. This type of address is not routable outside the local network. The Global ID and the Subnet ID (SID) are allocated by the network administrator;

– *Link Local Unicast* is a unicast address allowed for a link local (without router). This type of address is not routable outside a local link;

– *Multicast* is an address assigned to different devices of the network. Following the scope (SC) of the address, the multicast group may be either Interface-local, Link-local, Admin-local, Site-local, Organization-local or global. For more information about Flag and SC (scope) parameters, see RFC 4291, 2.7.

It is important to note that there is no broadcast address defined in IPv6.

For more information concerning IPv6 addressing, see RFC 4291.

C.3 IPv6 header format

As defined in RFC 2460, Clause 3, the header of one IPv6 packet is composed as shown in Figure C.2:

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

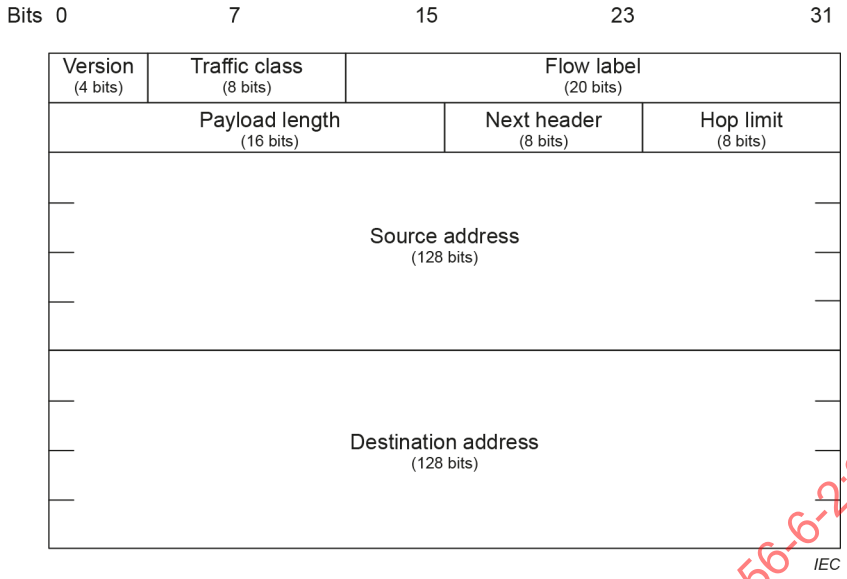


Figure C.2 – IPv6 header format

Where:

- *Version* specifies the version of the protocol. For IPv6, the value is fixed and equals to 6;
- *Traffic class* is used by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets (see RFC 2474, Clause 3). Note that the traffic class value is only a notion of prioritization used to distribute the IPv6 frame and do not secure the transmission (the philosophy of the IP network is to do its best).

  Figure C.3 shows the content of the traffic class parameter:

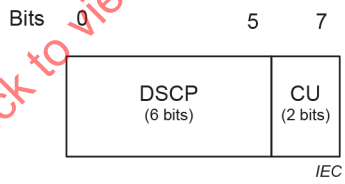


Figure C.3 – Traffic class parameter format

  Where:

  – DSCP – Differentiated services code point contains the prioritization of the IPv6 packet in the network (see RFC 2474 for details);

  – CU – Currently unused;

- *Flow label* may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. Currently, no fully definition of this field is given by IETF and may be seen as reserved for future used;
- *Payload length* indicates the size of the upper layer payload carried by the IPv6 frame;
- *Next header* identifies the type of header immediately following the IPv6 header of the current frame. It may indicate an upper applicative header (ICMP, UDP, TCP, etc.) or extensions;
- *Hop limit* defines the time of life of the IPv6 packet in term of number of hop. The usage is similar to the one defined for IPv4 (decremented by 1 by each node that forwards the packet, the packet is discarded if Hop Limit is decremented to zero);
- *Source and destination addresses* indicate the originator of the IPv6 packet and the intended recipient.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Table C.1 summarizes the IPv6 headers managed / not managed by the IPv6 objects.

Table C.1 – IPv6 header vs. IPv6 IC

Parameters	IPv6 setup IC
Version	Not managed
Traffic class	Managed
Flow label	Not managed
Payload length	Not managed
Next header	Not managed
Hop limit	Not managed
Source and destination addresses	Not managed

C.4 IPv6 header extensions

C.4.1 Overview

Contrary to IPv4, IPv6 provides an extensible header by adding optional elements one by one. Currently, the following list of optional headers is defined by RFC 2460, see Table C.2.

Table C.2 summarizes the options managed / not managed by the IPv6 setup objects.

Table C.2 – Optional IPv6 header extensions vs. IPv6 IC

Extensions	IPv6 setup IC	See subclause
Hop-by-Hop options	Not managed	C.4.2
Destination options	Not managed	C.4.3
Routing options	Not managed	C.4.4
Fragment options	Not managed	C.4.5
Security options (Authentication and Encapsulating Security Payload)	Not managed	C.4.6
NOTE The options are listed in the order as they appear in the packet.		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

C.4.2 Hop-by-Hop options

The Hop-by-Hop options field shall be examined by all devices on the path. Four elements currently defined may compose this header (see RFC 2460, 4.3):

- A padding form Pad1 which introduces one byte of padding (see RFC 2460, 4.2);
- A padding form PadN which introduces more than 2 bytes of padding (see RFC 2460, 4.2);
- A Jumbogram field to indicate the length of the IPv6 datagram in case of jumbo payload (higher than the maximum size of length field of IPv6 header) (see RFC 2460 and RFC 2675); and
- A router alert: The option indicates that the contents of the datagram may be interesting to the router (see RFC 2711).

These optional elements are directly managed by the IPv6 protocol stack. Therefore, they are not managed by the COSEM IPv6 setup class.

C.4.3 Destination options

The Destination options field shall be examined only by the target device of the IP datagram. Four elements are currently defined by IETF (see RFC 2460, 4.6):

- A padding form Pad1 which introduces one byte of padding (see RFC 2460, 4.2);
- A padding form PadN which introduces more than 2 bytes of padding (see RFC 2460, 4.2);
- Mobility parameters (linked to new IP wireless networks – UMTS, LTE, etc.) (see RFC 3775); and
- Tunnelling options which permit to communicate between two IPv6 networks separated by an IPv4 network (6To4 mechanism) (see RFC 2460).

NOTE The Mobility parameters and the Tunnelling options have been defined separately from RFC 2460 and are not mentioned in this document. See appropriate RFCs for more details.

These optional elements are directly managed by the IPv6 protocol stack. These optional fields are not managed by the COSEM IPv6 setup class.

C.4.4 Routing options

Routing options element is used to specify some routing parameters (see RFC 2460, 4.4). Currently, only one type is defined by IETF (type 2), which is used in case of mobility IPv6.

This notion is out of scope of the COSEM IPv6 setup class.

For information, due to an important security issue (see RFC 5095), the type 0 previously defined has been depreciated and is forbidden. Furthermore, the type 1 has been temporarily defined for experimental Nimrod and is obsolete since 1996.

C.4.5 Fragment options

Fragment options field is used in the case of fragmentation of the applicative payload if its size is higher than the MTU of the network (see RFC 2460, 4.5).

This element is directly managed by the IPv6 protocol stack. These optional fields are not managed by the COSEM IPv6 setup class.

C.4.6 Security options

Contrary to IPv4, IPv6 protocol layer includes natively the IPSec protocol. These extensions are fully defined in the RFC 4302 and RFC 4303.

The security options are composed of two extensible headers:

- Authentication Header (AH): Contains information used to verify the authenticity of most parts of the packet (see RFC 4302, Clause 2);
- Encapsulating Security Payload (ESP): Carries encrypted data for secure communication (see RFC 4303, Clause 2).

Due to the complexity of the IPsec protocol, the configuration of IPSec is out of scope of this document and shall be treated separately.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Annex D
(informative)

Overview of the narrow-band OFDM PLC technology for PRIME networks

For the specification of the PRIME narrow-band OFDM PLC setup classes, see 4.12.

NOTE This technology is supported by the PRIME Alliance, http://www.prime-alliance.org.

ITU-T G.9904:2012 specifies a physical layer, a medium access control layer and convergence layers for cost-effective narrowband (<200 kbps) data transmission over electrical power lines, intended for use in smart metering and smart grid applications. It is based on Orthogonal Frequency Division Multiplexing (OFDM).

The specification currently describes the following:

- a low-cost PHY capable of achieving rates of encoded 128 kbps;
- a Master-Slave MAC optimised for the power line environment;
- a convergence layer for the LLC layer specified in IEC 61334-4-32;
- a convergence layer for IPv4;
- a convergence layer for IPv6;

The DLMS®/COSEM communication profiles for narrow-band OFDM PLC PRIME neighbourhood networks are specified in IEC 62056-8-4.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Annex E
(informative)

Overview of the narrow-band OFDM PLC technology for G3-PLC networks

For the specification of the G3 narrow-band OFDM PLC setup classes, see 4.13.

NOTE This specification is supported by the G3-PLC Alliance, http://www.G3-PLC.com.

ITU-T G.9903 Amd. 1:2017 specifies the physical, MAC and 6LoWPAN Adaptation layers of the G3-PLC technology while ITU-T G.9901:2014 deals with frequency bandplan allocation and associated transmission level limitations.

Power line communication has been used for many decades, but a variety of new services and applications require more reliability and higher data rates. However, the power line channel is very hostile. Channel characteristics and parameters vary with frequency, location, time and the type of equipment connected to it. The lower frequency regions from 10 kHz to 200 kHz are especially susceptible to interference. Furthermore, the power line is a very frequency selective channel. Besides background noise, it is subject to impulsive noise often occurring at 50/60 Hz and group delays up to several hundred microseconds.

G3-PLC uses advanced modulation and channel coding techniques, which enables efficient use of the limited bandwidth of the CENELEC bands and facilitates communication over the power line channel. This combination enables a very robust communication in the presence of narrow-band interference, impulsive noise, and frequency selective attenuation. The specification addresses the following main objectives:

- provide robust communication on extremely harsh power line channels;
- provide a minimum of 20 kbps effective data rate in the normal mode of operation;
- ability to notch selected frequencies, to allow the cohabitation with other Narrow-band PLC communication technologies (e.g. IEC 61334-5-1:2001 S-FSK) or to be compliant with specific regulatory requirements;
- dynamic tone adaptation capability to select frequencies on the channel that do not have major interference, thereby ensuring a robust communication;
- access control, authentication, confidentiality and integrity to ensure high level of security.

To this end, the G3-PLC protocol stack aggregates several layers and sub-layers that form the G3-PLC profile:

- a robust high-performance PHY layer based on OFDM and adapted to the narrow-band PLC environment;
- a MAC layer of the IEEE 802.15.4:2006 type (extended), well suited to low data rates;
- IPv6, the new generation of IP (Internet Protocol), which widely opens the range of potential applications and services; and
- to allow good IPv6 and MAC interoperability, an Adaptation sublayer taken from the Internet world (IETF) and called 6LoWPAN (RFC 4944 extended and RFC 6282). The adaptation sub layer also embeds the LOADng routing algorithm to allow multi-hop mesh connectivity.

For more information about the extensions of IEEE 802.15.4:2006, RFC 4944 and RFC 6282 (AKA 6LoWPAN) standards, and LOADng, see ITU-T G.9903 Amd. 1:2017.

The DLMS®/COSEM narrow-band G3-PLC communication profile is specified in IEC 62056-8-5:2017:2017.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Annex F
(informative)

Significant technical changes with respect to IEC 62056-6-2, Edition 3.0:2017

This Edition 4.0 of IEC 62056-6-2 includes the following significant technical changes with respect to IEC 62056-6-2, Ed.3.0:2017.

Item	Subclause	Description
1.	4.4.8.2	Class updated – 40 – Push set up
2.	4.4.12	New class added – 124 – Communication port protection
3.	4.5.10	Class updated – 65 – Parameter monitor
4.	4.7.8	Class updated – 47 – GSM diagnostic
5.	4.7.9	Calss updated – 151 – LTE monitoring
6.	4.13.5	Class updated – 92 – G3-PLC 6LoWPAN
7.	4.16	New interface classes for LPWAN.
8.	4.16.3	New interface classes for LoRaWAN
9.	4.17	New interface classes for WiSUN.
10.	4.18	New interface classes for ISO/IEC 14908 PLC networks.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Bibliography

ANSI/TIA-4957.200, *Layer 2 Standard Specification for the Smart Utility Network*

IEC 61334-6:2000, *Distribution automation using distribution line carrier systems – Part 6: A-XDR encoding rule*

IEC 62055-41, *Electricity metering – Payment systems – Part 41: Standard transfer specification (STS) – Application layer protocol for one-way token carrier systems*

IEC 62056-8-5:2017:, *Electricity metering data exchange –The DLMS®/COSEM suite – Part 8-5: Narrow-band OFDM G3-PLC communication profile for neighbourhood networks*

ITU-T G.9901:2014, *Series G: Transmission systems and media, digital systems and networks – Access Networks – In premises networks – Narrow-band orthogonal frequency division multiplexing power line communication transceivers – Power spectral density specification*

ITU Recommendation X.217:1995, *Information technology – Open Systems Interconnection – Service definition for the association control service element*

ITU Recommendation X.227:1995, *Information technology – Open Systems Interconnection Connection-oriented protocol for the association control service element: Protocol specification*

IETF STD 5, *Internet Protocol,* 1981. (Also IETF RFC 0791, RFC 0792, RFC 0919, RFC 0922, RFC 0950, RFC 1112)

IETF STD 51 / RFC 1661, *The Point-to-Point Protocol (PPP)* (Also: IETF STD 0051), 1994, Updated by: RFC 2153, Obsoletes: RFC 1548

IETF STD 51 / RFC 1662, *PPP in HDLC-like Framing*, (Also: IETF STD 0051), 1994, Obsoletes: RFC 1549

3GPP TS 36.214 V13.0.0 (2015-12), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer; Measurements*

The following RFCs are available online from the Internet Engineering Task Force (IETF): http://www.ietf.org/rfc/std-index.txt, http://www.ietf.org/rfc/

RFC 768, *User Datagram Protocol*

RFC 791, *Internet Protocol* (Also: IETF STD 0005), 1981. RFC 1332, *The PPP Internet Protocol Control Protocol* (IPCP), 1992, Updated by: RFC 3241. Obsoletes: RFC 1172.

RFC 793, *Transmission Control Protocol (Also IETF STD 0007),* 1981, Updated by: RFC 3168

RFC 940, *Toward an Internet Standard Scheme for Subnetting,* 1985

RFC 950, *Internet Standard Subnetting Procedure,* 1985

RFC 1144, *Compressing TCP/IP Headers for Low-Speed Serial Links,* 1990

RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

RFC 1332, *The PPP Internet Protocol Control Protocol (IPCP), 1992, Updated by: RFC 3241. Obsoletes: RFC 1172*

RFC 1570, *PPP LCP Extensions*, 1994

RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP),* 1996. Obsoletes: RFC 1334

RFC 2433, *PPP CHAP Extension*, 1998

RFC 2460, *Internet Protocol, Version 6 (IPv6),* 1998

RFC 2473, *Generic Packet Tunneling in IPv6,* 1998

RFC 2474, *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*, 1998

RFC 2507, *IP Header Compression*, 1999

RFC 2508, *Compressing IP/UDP/RTP Headers for Low-Speed Serial Links,* 1999

RFC 2675, *IPv6 Jumbograms,* 1999

RFC 2711, *IPv6 Router Alert Option,* 1999

RFC 2759, *Microsoft PPP CHAP Extensions*, Version 2, 2000

RFC 2986, *PKCS #10 v1.7: Certification Request Syntax Standard*

RFC 3095, *RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed,* 2001

RFC 3241, *Robust Header Compression (ROHC) over PPP,* 2002. Updates: RFC1332

RFC 3315 *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)*

RFC 3513, *Internet Protocol Version 6 (IPv6) Addressing Architecture,* 2003

RFC 3544, *IP Header Compression over PPP*, 2003

RFC 3748, *Extensible Authentication Protocol (EAP)*, 2004

RFC 3775, *Mobility Support in IPv6,* 2004

RFC 4291, *IP Version 6 Addressing Architecture*

RFC 4302, *IP Authentication Header,* 2005

RFC 4303, *IP Encapsulating Security Payload (ESP),* 2005

RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

RFC 4861, *Neighbor Discovery for IP version 6 (IPv6),* 2007

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

RFC 4944, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks,* 2007

RFC 5095, *Deprecation of Type 0 Routing Headers in IPv6,* 2007

RFC 5216, *The EAP-TLS Authentication Protocol*

RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,* 2008

RFC 5905, *Network Time Protocol Version 4: Protocol and Algorithms Specification,* 2010

RFC 6206, *The Trickle Algorithm*

RFC 6282, *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks* [online]. Edited by J. Hui, Ed. September 2011

RFC 6550, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*

RFC 6775, *Neighbor Discovery Optimization for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs),* 2012

RFC 7731, *Multicast Protocol for Low-Power and Lossy Networks (MPL)*

RFC 7774, *Multicast Protocol for Low-Power and Lossy Networks (MPL) Parameter Configuration Option for DHCPv6*

Point-to-Point (PPP) Protocol Field Assignments. *Online database*. Available from: http://www.iana.org/assignments/ppp-numbers/ppp-numbers.xhtml

DLMS UA 1000-1, the “Blue Book” Ed. 12.1:2015, *COSEM interface classes and OBIS identification system*

DLMS UA 1000-2, the “Green Book” Ed. 8.1:2015, *DLMS/COSEM Architecture and Protocols*

DLMS UA 1001-1, the “Yellow Book”, Ed. 5.0:2015, *DLMS/COSEM Conformance test and certification process*

DLMS UA 1002, the “White Book”, Ed. 1.0:2003, *COSEM Glossary of terms*

3GPP TS 24.008 V13.7.0 (2016-10), *Technical Specification Digital cellular telecommunications system (Phase 2+) (GSM); Universal Mobile Telecommunications System (UMTS); LTE; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3*

3GPP TS 24.301 V13.4.0 (2016-01), *Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS); Stage 3*

3GPP TS 24.301 V13.11.0 (2018-01), *Technical Specification Group Core Network and Terminals; Non-Access-Stratum (NAS) protocol for Evolved Packet System (EPS);Stage 3*

3GPP TS 27.007 V16.1.0 (2019-06) *Technical Specification Group Core Network and Terminals; AT command set for User Equipment (UE)*

3GPP TS 36.101 V15.4.0 (2019-01), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3GPP TS 36.133 V13.11.0 (2018-04), *Technical Specification LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management*

3GPP TS 36.133 V14.4.0 (2017-07), *Technical Specification LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Requirements for support of radio resource management*

3GPP TS 36.213 V15.5.0 (2019-05), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures*

3GPP TS 36.304 V13.0.0 (2015-12), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode*

3GPP TS 36.304 V13.8.0 (2018-01), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) procedures in idle mode*

3GPP TS 36.321 V15.5.0 (2019-05), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification*

3GPP TS 36.331 V15.5.1 (2019-05), *Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*

ETSI GSM 05.08:1996, *Digital cellular telecommunications system (Phase 2+); Radio subsystem link control*

ANSI C12.19:1997, IEEE 1377:1997, *Utility industry end device data tables*

DIN 43863-3:1997, *Tariff metering device as additional equipment for electricity meters – EDIS – Energy Data Identification System*

ZigBee® 053474 ZigBee® Specification. *The specification can be downloaded free of charge from*
http://www.zigbee.org/Standards/ZigBeeSmartEnergy/Specification.aspx

[FANSPEC], *Wi-SUN Alliance: Field Area Network Working Group (FANWG):Technical Profile Specification:Field Area Network:Version 1v26*

[PHYSPEC], *Wi-SUN Alliance: PHY Working Group (PHYWG) Wi-SUN PHY Specification Revision 1V02*

LoRaWAN *1.0.3, LoRaWAN® Specification v1.0.3*

https://lora-alliance.org/resource-hub/lorawanr-specification-v103

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

SOMMAIRE

AVANT-PROPOS 538
INTRODUCTION 540
1 Domaine d'application 542
2 Références normatives 542
3 Termes, définitions et abréviations 545
3.1 Termes et définitions liés au processus de transfert d'image (voir 4.4.6) 545
3.2 Termes et définitions liés aux classes "S-FSK PLC setup" (voir 4.10) 546
3.3 Termes et définitions liés aux classes d'interfaces "PRIME NB OFDM PLC setup" (voir 4.12) 547
3.4 Termes et définitions liés aux classes d'interfaces "ISO/IEC 14908 setup" (voir 4.18) 549
3.5 Termes et définitions liés à ZigBee® (voir 4.15) 549
3.6 Termes et définitions liés aux classes d'interfaces de comptage à paiement (voir 4.6) 551
3.7 Termes et définitions liés à la classe d'interfaces Arbitrator (voir 0) 555
3.8 Abréviations 555
4 Classes d'interfaces COSEM 560
4.1 Principes de base 560
4.1.1 Généralités 560
4.1.2 Méthodes de référencement 562
4.1.3 Base_name réservés pour des objets COSEM spéciaux 562
4.1.4 Notation pour la description des classes 563
4.1.5 Types de données communs 566
4.1.6 Formats de données 567
4.1.7 Modèle de serveur COSEM 573
4.1.8 Dispositif logique COSEM 573
4.1.9 Sécurité des informations 575
4.2 Vue d'ensemble des classes d'interfaces COSEM 575
4.3 Classes d'interfaces pour les paramètres et les données de mesure 581
4.3.1 Data (class_id = 1, version = 0) 581
4.3.2 Register (class_id = 3, version = 0) 582
4.3.3 Extended register (class_id = 4, version = 0) 587
4.3.4 Demand register (class_id = 5, version = 0) 588
4.3.5 Register activation (class_id = 6, version = 0) 592
4.3.6 Profile generic (class_id = 7, version = 1) 594
4.3.7 Utility tables (class_id = 26, version = 0) 600
4.3.8 Register table (class_id = 61, version = 0) 601
4.3.9 Status mapping (class_id = 63, version = 0) 604
4.3.10 Compact data (class_id: 62, version: 1) 605
4.4 Classes d'interfaces pour le contrôle et la gestion des accès 613
4.4.1 Vue d'ensemble 613
4.4.2 Identification des utilisateurs du client 613
4.4.3 Association SN (class_id = 12, version = 4) 614
4.4.4 Association LN (class_id = 15, version = 3) 619
4.4.5 SAP assignment (class_id = 17, version = 0) 626
4.4.6 Image transfer (class_id = 18, version = 0) 627
4.4.7 Security setup (class_id = 64, version = 1) 635

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8 Classes d'interfaces et objets Push 641
4.4.9 COSEM data protection (class_id = 30, version = 0) 655
4.4.10 Function control (class_id: 122, version: 0) 670
4.4.11 Array manager (class_id = 123, version = 0) 672
4.4.12 Communication port protection (class_id = 124, version = 0) 677
4.5 Classes d'interfaces pour commande à limite temporelle et événementielle 680
4.5.1 Clock (class_id = 8, version = 0) 680
4.5.2 Script table (class_id = 9, version = 0) 684
4.5.3 Schedule (class_id = 10, version = 0) 685
4.5.4 Special days table (class_id = 11, version = 0) 689
4.5.5 Activity calendar (class_id = 20, version = 0) 690
4.5.6 Register monitor (class_id = 21, version = 0) 695
4.5.7 Single action schedule (class_id = 22, version = 0) 696
4.5.8 Disconnect control (class_id = 70, version = 0) 697
4.5.9 Limiter (class_id = 71, version = 0) 701
4.5.10 Parameter monitor (class_id = 65, version = 1) 703
4.5.11 Sensor manager (class_id = 67, version = 0) 707
4.5.12 Arbitrator (class_id = 68, version = 0) 711
4.5.13 Exemples de modélisation: tarification et facturation 715
4.6 Classes d'interfaces relatives au comptage à paiement 717
4.6.1 Présentation du modèle comptable COSEM 717
4.6.2 Account (class_id = 111, version = 0) 719
4.6.3 Credit interface class (class_id = 112, version = 0) 728
4.6.4 Charge (class_id = 113, version = 0) 739
4.6.5 Token gateway (class_id = 115, version = 0) 745
4.7 Classes d'interfaces pour l'établissement d'échange de données via les ports locaux et les modems 747
4.7.1 IEC local port setup (class_id = 19, version = 1) 747
4.7.2 IEC HDLC setup (class_id = 23, version = 1) 749
4.7.3 IEC twisted pair (1) setup (class_id = 24, version = 1) 751
4.7.4 Modem configuration (class_id = 27, version = 1) 755
4.7.5 Auto answer (class_id = 28, version = 2) 756
4.7.6 Auto connect (class_id = 29, version = 2) 760
4.7.7 GPRS modem setup (class_id = 45, version = 0) 762
4.7.8 GSM diagnostic (class_id: 47, version: 2) 763
4.7.9 LTE monitoring (class_id: 151, version: 1) 765
4.8 Classes d'interfaces pour l'établissement d'échange de données via M-Bus 768
4.8.1 Vue d'ensemble 768
4.8.2 M-Bus slave port setup (class_id = 25, version = 0) 769
4.8.3 M-Bus client (class_id = 72, version = 1) 770
4.8.4 Wireless Mode Q channel (class_id = 73, version = 1) 774
4.8.5 M-Bus master port setup (class_id = 74, version = 0) 775
4.8.6 DLMS®/COSEM server M-Bus port setup (class_id = 76, version = 0) 776
4.8.7 M-Bus diagnostic (class_id = 77, version = 0) 778
4.9 Classes d'interfaces pour l'établissement d'échange de données sur Internet 781
4.9.1 TCP-UDP setup (class_id = 41, version = 0) 781
4.9.2 IPv4 setup (class_id = 42, version = 0) 782
4.9.3 IPv6 setup (class_id = 48, version = 0) 786
4.9.4 MAC address setup (class_id = 43, version = 0) 790

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.5 PPP setup (class_id = 44, version = 0) 790
4.9.6 SMTP setup (class_id = 46, version = 0) 795
4.9.7 NTP setup (class_id = 100, version = 0) 796
4.10 Classes d'interfaces pour l'établissement d'échange de données en utilisant le PLC à modulation S-FSK 798
4.10.1 Généralités 798
4.10.2 Vue d'ensemble 798
4.10.3 S-FSK Phy&MAC set-up (class_id = 50, version = 1) 801
4.10.4 S-FSK Active initiator (class_id = 51, version = 0) 806
4.10.5 S-FSK MAC synchronization timeouts (class_id = 52, version = 0) 808
4.10.6 S-FSK MAC counters (class_id = 53, version = 0) 810
4.10.7 IEC 61334-4-32 LLC setup (class_id = 55, version = 1) 814
4.10.8 S-FSK Reporting system list (class_id = 56, version = 0) 815
4.11 Classes d'interfaces pour l'établissement de la couche LLC selon l'ISO/IEC 8802-2 815
4.11.1 Généralités 815
4.11.2 ISO/IEC 8802-2 LLC Type 1 setup (class_id = 57, version = 0) 816
4.11.3 ISO/IEC 8802-2 LLC Type 2 setup (class_id = 58, version = 0) 816
4.11.4 ISO/IEC 8802-2 LLC Type 3 setup (class_id = 59, version = 0) 818
4.12 Classes d'interfaces pour l'établissement et la gestion du profil PLC OFDM à bande étroite DLMS®/COSEM pour les réseaux PRIME 820
4.12.1 Vue d'ensemble 820
4.12.2 Mise en correspondance des attributs PRIME NB OFDM PLC PIB avec les attributs d'IC COSEM 821
4.12.3 61334-4-32 LLC SSCS setup (class_id = 80, version = 0) 823
4.12.4 Paramètres de couche physique PRIME NB OFDM PLC 824
4.12.5 PRIME NB OFDM PLC Physical layer counters (class_id = 81, version = 0) 824
4.12.6 PRIME NB OFDM PLC MAC setup (class_id = 82, version = 0) 825
4.12.7 NB OFDM PLC MAC functional parameters (class_id = 83 version = 0) 827
4.12.8 PRIME NB OFDM PLC MAC counters (class_id = 84, version = 0) 829
4.12.9 PRIME NB OFDM PLC MAC network administration data (class_id = 85, version = 0) 830
4.12.10 PRIME NB OFDM PLC MAC address setup (class_id = 43, version = 0) 833
4.12.11 PRIME NB OFDM PLC Application identification (class_id = 86, version = 0) 833
4.13 Classes d'interfaces pour l'établissement et la gestion du profil PLC OFDM à bande étroite DLMS®/COSEM pour les réseaux G3-PLC 834
4.13.1 Vue d'ensemble 834
4.13.2 Mise en correspondance des attributs G3-PLC PIB avec les attributs d'IC COSEM 835
4.13.3 G3-PLC MAC layer counters (class_id = 90, version = 1) 836
4.13.4 G3-PLC MAC setup (class_id = 91, version = 2) 838
4.13.5 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 2) 846
4.14 Classes d'interfaces pour l'établissement et la gestion des réseaux de voisinage DLMS®/COSEM HS-PLC selon l'ISO/IEC 12139-1 853
4.14.1 Vue d'ensemble 853
4.14.2 HS-PLC ISO/IEC 12139-1 MAC setup (class_id = 140, version = 0) 853
4.14.3 HS-PLC ISO/IEC 12139-1 CPAS setup (class_id = 141, version = 0) 855
4.14.4 HS-PLC ISO/IEC 12139-1 IP SSAS setup (class_id = 142, version = 0) 856

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.14.5 HS-PLC ISO/IEC 12139-1 HDLC SSAS setup (class_id = 143, version = 0) 856
4.15 Classes d'établissement ZigBee® 857
4.15.1 Vue d'ensemble 857
4.15.2 ZigBee® SAS startup (class_id = 101, version = 0) 859
4.15.3 ZigBee® SAS join (class_id = 102, version = 0) 861
4.15.4 ZigBee® SAS APS fragmentation (class_id = 103, version = 0) 863
4.15.5 ZigBee® network control (class_id = 104, version = 0) 863
4.15.6 ZigBee® tunnel setup (class_id = 105, version = 0) 869
4.16 Classes d'interfaces pour l'établissement et la gestion du profil DLMS®/COSEM pour les réseaux LPWAN 871
4.16.1 Généralités 871
4.16.2 Classes d'interfaces génériques 871
4.16.3 Classes d'interfaces spécifiques LPWAN 876
4.17 Classes d'interfaces pour l'établissement et la gestion du profil DLMS®/COSEM pour les réseaux Wi-SUN 883
4.17.1 Wi-SUN setup (class_id = 95, version 0) 883
4.17.2 Wi-SUN diagnostic (class_id = 96, version 0) 888
4.17.3 RPL diagnostic (class_id = 97, version 0) 891
4.17.4 MPL diagnostic (class_id = 98, version 0) 893
4.18 Classes d'interfaces pour l'établissement et la gestion du profil DLMS®/COSEM pour les réseaux PLC selon l'ISO/IEC 14908 896
4.18.1 Généralités 896
4.18.2 ISO/IEC 14908 identification (class_id = 130, version = 0) 896
4.18.3 ISO/IEC 14908 protocol setup (class_id = 131, version = 0) 897
4.18.4 ISO/IEC 14908 protocol status (class_id = 132, version = 0) 897
4.18.5 ISO/IEC 14908 diagnostic (class_id = 133, version = 0) 899
5 Précédentes versions des classes d'interfaces 902
5.1 Généralités 902
5.1.1 Nouvelles versions des classes d'interfaces 902
5.1.2 Nouvelles classes d'interfaces 902
5.1.3 Retrait de classes d'interfaces 902
5.2 Versions précédentes des classes d'interfaces – Généralités 902
5.3 Précédentes classes d'interfaces pour les paramètres et données de mesure 902
5.3.1 Profile generic (class_id = 7, version = 0) 902
5.3.2 Compact data (class_id = 62, version = 0) 906
5.4 Précédentes versions des classes d'interfaces pour le contrôle et la gestion des accès 909
5.4.1 Association SN (class_id = 12, version = 0) 909
5.4.2 Association SN (class_id = 12, version = 1) 911
5.4.3 Association SN (class_id = 12, version = 2) 914
5.4.4 Association SN (Class_id = 12, version =3) 918
5.4.5 Association LN (class_id = 15, version = 0) 923
5.4.6 Association LN (class_id = 15, version = 1) 928
5.4.7 Association LN (class_id = 15, version = 2) 933
5.4.8 Security setup (class_id = 64, version = 0) 939
5.4.9 Push Setup (class_id = 40, version = 0) 941
5.4.10 Push Setup (class_id = 40, version = 1) 946
5.5 Précédentes versions des classes d'interfaces pour commande à limite temporelle et événementielle 953

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

5.5.1 Parameter monitor (class_id = 65, version = 0) 953
5.6 Versions précédentes des classes d'interfaces liées aux compteurs de paiement 955
5.7 Précédentes versions des classes d'interfaces pour l'établissement d'échange de données via les ports locaux et les modems 955
5.7.1 IEC local port setup (class_id = 19, version = 0) 955
5.7.2 IEC HDLC setup, (class_id = 23, version = 0) 957
5.7.3 IEC twisted pair (1) setup (class_id = 24, version = 0) 959
5.7.4 PSTN modem configuration (class_id = 27, version = 0) 961
5.7.5 Auto answer (class_id = 28, version = 0) 962
5.7.6 PSTN auto dial (class_id = 29, version = 0) 964
5.7.7 Auto connect (class_id = 29, version = 1) 966
5.7.8 GSM diagnostic (class_id = 47, version = 0) 967
5.7.9 GSM diagnostic (class_id: 47, version: 1) 969
5.7.10 LTE monitoring (class_id: 151, version: 0) 972
5.8 Précédentes versions des classes d'interfaces pour l'établissement d'échange de données via M-Bus 973
5.8.1 M-Bus client (class_id = 72, version = 0) 973
5.9 Précédentes versions des classes d'interfaces pour l'établissement d'échange de données sur Internet 978
5.10 Précédentes versions des classes d'interfaces pour l'échange de données utilisant S-FSK PLC 978
5.10.1 S-FSK Phy&MAC setup (class_id = 50, version = 0) 978
5.10.2 S-FSK IEC 61334-4-32 LLC setup (class_id = 55, version = 0) 983
5.11 Précédentes versions des classes d'interfaces pour l'établissement de la couche LLC pour l'ISO/IEC 8802-2 984
5.12 Précédentes versions des classes d'interfaces pour l'établissement et la gestion du profil PLC OFDM à bande étroite DLMS®/COSEM pour les réseaux PRIME 984
5.13 Précédentes versions des classes d'interfaces pour l'établissement et la gestion du profil PLC OFDM à bande étroite DLMS®/COSEM pour les réseaux G3-PLC 984
5.13.1 Mise en correspondance des attributs PIB G3-PLC avec les attributs d'IC COSEM (version d'origine) 984
5.13.2 G3 NB OFDM PLC MAC layer counters (class_id = 90, version = 0) 986
5.13.3 G3 NB OFDM PLC MAC setup (class_id = 91, version = 0) 987
5.13.4 G3 NB OFDM PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 0) 991
5.13.5 Mise en correspondance des attributs PIB G3-PLC avec les attributs d'IC COSEM (version précédente) 997
5.13.6 G3-PLC MAC setup (class_id = 91, version = 1) 999
5.13.7 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 1) 1005
5.14 Précédentes versions des classes d'interfaces pour l'établissement et la gestion des réseaux de voisinage DLMS®/COSEM HS-PLC selon l'ISO/IEC 12139-1 1011
5.15 Précédentes versions des classes d'établissement ZigBee® 1011
6 Relation à l'OBIS 1011
6.1 Généralités 1011
6.2 Objets COSEM abstraits 1012
6.2.1 Utilisation du groupe de valeurs C 1012
6.2.2 Données relatives aux périodes de facturation historiques 1013

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.3 Valeurs des périodes de facturation / réinitialisation d'entrées de compteur 1015
6.2.4 Autres codes OBIS abstraits à usage général 1015
6.2.5 Objets "Clock" (class_id = 8) 1016
6.2.6 Configuration de modem et objets connexes 1016
6.2.7 Objets "Script table" (class_id = 9) 1017
6.2.8 Objets "Special days table" (class_id = 11) 1018
6.2.9 Objets "Schedule" (class_id = 10) 1018
6.2.10 Objets "Activity calendar" (class_id = 20) 1019
6.2.11 Objets "Register activation" (class_id = 6) 1019
6.2.12 Objets "Single action schedule" (class_id = 22) 1019
6.2.13 Objets "Register monitor" (class_id = 21) 1020
6.2.14 Objets "Parameter monitor" (class_id = 65) 1020
6.2.15 Objets "Limiter" (class_id = 71) 1020
6.2.16 Objets "Array manager" (class_id = 123) 1020
6.2.17 Objets liés au comptage à paiement 1020
6.2.18 Objets "IEC local port setup" (class_id = 19) 1021
6.2.19 Objets "Standard readout profile" (class_id = 7) 1021
6.2.20 Objets "IEC HDLC setup" (class_id = 23) 1022
6.2.21 Objets "IEC twisted pair (1) setup" (class_id = 24) 1022
6.2.22 Objets liés à l'échange de données sur M-Bus 1023
6.2.23 Objets pour l'établissement d'échange de données sur Internet 1024
6.2.24 Objets "Push Setup" (class_id = 40) 1025
6.2.25 Objets pour l'établissement d'échange de données en utilisant PLC S-FSK 1026
6.2.26 Objets pour l'établissement de la couche LLC selon l'ISO/IEC 8802-2 1026
6.2.27 Objets pour l'échange de données à l'aide de l'OFDM PLC à bande étroite pour les réseaux PRIME 1027
6.2.28 Objets pour l'échange de données en utilisant l'OFDM PLC à bande étroite pour les réseaux G3-PLC 1028
6.2.29 Objets d'établissement ZigBee® 1028
6.2.30 Objets pour l'établissement et la gestion de l'échange de données en utilisant les réseaux PLC selon l'ISO/IEC 14908 1029
6.2.31 Objets pour l'échange de données en utilisant les réseaux HS-PLC selon l'ISO/IEC 12139-1 1029
6.2.32 Objets pour l'échange de données en utilisant les réseaux Wi-SUN 1030
6.2.33 Objets "Association" (class_id = 12, 15) 1030
6.2.34 Object "SAP assignment" (class_id = 17) 1030
6.2.35 Objet "COSEM logical device name" 1031
6.2.36 Objets relatifs à la sécurité des informations 1031
6.2.37 Objets "Image transfer" (class_id = 18) 1031
6.2.38 Objets "Function control" (class_id = 122) 1032
6.2.39 Objets "Communication port protection" (class_id = 124) 1032
6.2.40 Objets "Utility table" (class_id = 26) 1032
6.2.41 Objets "Compact data" (class_id = 62) 1033
6.2.42 Objets "Device ID" 1033
6.2.43 Objets "Metering point ID" 1034
6.2.44 Objets "Parameter changes" et "calibration" 1034
6.2.45 Objets "I/O control signal" 1034
6.2.46 Objets "Disconnect control" (class_id = 70) 1035

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

6.2.47 Objets "Arbitrator" (class_id = 68) 1035
6.2.48 Objets "Status of internal control signals" 1035
6.2.49 Objets "Internal operating status" 1036
6.2.50 Objets "Battery entries" 1036
6.2.51 Objets "Power failure monitoring" 1037
6.2.52 Objets "Operating time" 1037
6.2.53 Objets "Environment related parameters" 1037
6.2.54 Objets "Status register" 1038
6.2.55 Objets "Event code" 1038
6.2.56 Objets "Communication port log parameter" 1038
6.2.57 Objets "Consumer message" 1039
6.2.58 Objets "Currently active tariff" 1039
6.2.59 Objets "Event counter" 1039
6.2.60 Objets "Profile entry digital signature" 1039
6.2.61 Objets "Profile entry counter" 1040
6.2.62 Objets liés à des événements de fraude de compteur 1040
6.2.63 Objets "Error register" 1040
6.2.64 Objets "Alarm register", "Alarm filter" et "Alarm descriptor" 1041
6.2.65 Objets généraux sous forme de liste 1042
6.2.66 Objets Event log (class_id 7) 1042
6.2.67 Objets inactifs 1042
6.3 Objets COSEM liés à l'électricité 1043
6.3.1 Définitions du groupe de valeurs D 1043
6.3.2 Numéros d'ID – Électricité 1043
6.3.3 Valeurs des périodes de facturation/réinitialisation d'entrées de compteur 1044
6.3.4 Autres objets à usage général liés à l'électricité 1044
6.3.5 Algorithme de mesure 1045
6.3.6 ID de point de comptage (lié à l'électricité) 1047
6.3.7 Objets "Electricity related status" 1047
6.3.8 Objets "List" – Electricité (class_id = 7) 1048
6.3.9 Valeurs de seuil 1048
6.3.10 Objets "Register monitor" (class_id = 21) 1049
6.4 Codage des identifications OBIS 1050
Annexe A (informative) Informations supplémentaires relatives aux IC "Auto answer" et "Auto connect" 1051
Annexe B (informative) Informations supplémentaires relatives à M-Bus client (class_id = 72, version 1) 1053
Annexe C (informative) Informations supplémentaires relatives à la classe IPv6 setup (class_id = 48, version = 0) 1055
C.1 Généralités 1055
C.2 Adressage IPv6 1055
C.3 Format d'en-tête IPv6 1056
C.4 Extensions d'en-tête IPv6 1058
C.4.1 Vue d'ensemble 1058
C.4.2 Options Saut par saut 1058
C.4.3 Options Destination 1059
C.4.4 Options Routage 1059
C.4.5 Options Fragment 1059

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

C.4.6 Options Sécurité 1059
Annexe D (informative) Vue d'ensemble de la technologie OFDM PLC à bande étroite pour les réseaux PRIME 1060
Annexe E (informative) Vue d'ensemble de la technologie OFDM PLC à bande étroite pour les réseaux G3-PLC 1061
Annexe F (informative) Modifications techniques majeures par rapport à l'IEC 62056-6-2, Édition 3.0:2017 1063
Bibliographie 1064

Figure 1 – Signification des définitions concernant l'Image 546
Figure 2 – Classe d'interfaces et ses instances 561
Figure 3 – Modèle de serveur COSEM 573
Figure 4 – Dispositif de comptage combiné 573
Figure 5 – Vue d'ensemble des classes d'interfaces – Partie 1 576
Figure 6 – Vue d'ensemble des classes d'interfaces – Partie 2 577
Figure 8 – Attributs de temps pour mesurer une demande glissante 588
Figure 9 – Attributs dans le cas d'une demande en bloc 589
Figure 10 – Attributs dans le cas d'une demande glissante (nombre de périodes = 3) 589
Figure 11 – Organigramme du processus de transfert d'images 633
Figure 12 – Modèle COSEM d'opération Push 642
Figure 13 – Fenêtres d'envoi Push et retards 644
Figure 14 – Modèle COSEM de protection des données 656
Figure 15 – Exemple: lecture de l'attribut *protection_buffer* 658
Figure 16 – Exemple de gestion d'un tableau 673
Figure 17 – Concept de temps généralisé 681
Figure 18 – Diagramme des transitions d'état de l'IC "Disconnect control" 698
Figure 19 – Définition des seuils supérieurs et inférieurs 711
Figure 20 – Modèle de tarification COSEM (exemple) 716
Figure 21 – Modèle de facturation COSEM (exemple) 716
Figure 22 – Présentation du modèle comptable 718
Figure 23 – Schéma des relations entre les attributs 719
Figure 24 – États du Crédit lorsque la priorité > 0 729
Figure 25 – Fonctionnement des fanions current_credit_status 732
Figure 26 – Interaction de current_credit_amount et d'available_credit avec le "crédit" de jeton et le "crédit" d'urgence 738
Figure 27 – Modèle d'objet des serveurs DLMS®/COSEM 799
Figure 28 – Modèle d'objet des serveurs DLMS®/COSEM 820
Figure 29 – Exemple de réseau ZigBee® 858
Figure 30 – Fenêtres d'envoi Push et retards 942
Figure 31 – Données de périodes de facturation historiques – Exemple avec modulo 12, VZ = 5 1014
Figure A.1 – Exemple de connectivité d'un réseau GSM/GPRS 1051
Figure B.1 – Diagramme d'états de clé de cryptage 1053
Figure C.1 – Formats d'adresse IPv6 1056
Figure C.2 – Format d'en-tête IPv6 1057

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure C.3 – Format du paramètre Traffic class 1057

Tableau 1 – Base_name réservés pour le référencement SN 562
Tableau 2 – Vue d'ensemble de la classe d'interfaces 563
Tableau 3 – Types de données communs 566
Tableau 4 – Liste des classes d'interfaces par class_id 578
Tableau 5 – Valeurs énumérées pour les unités physiques 584
Tableau 6 – Exemples d'utilisation de scaler_unit 586
Tableau 7 – Paramètres pour l'accès sélectif à l'attribut buffer 598
Tableau 8 – Paramètres pour l'accès sélectif à l'attribut buffer 600
Tableau 9 – Données de facturation quotidiennes 609
Tableau 10 – Attributs de l'objet "Compact data" 609
Tableau 11 – Codage A-XDR des données (SEQUENCE DE Get-Data-Result) 610
Tableau 12 – Données de diagnostic et d'alarme 610
Tableau 13 – Attributs de l'objet "Compact data" 611
Tableau 14 – Codage des données lues dans l'attribut buffer d'un objet "Profile generic" 611
Tableau 15 – Données du journal de bord 611
Tableau 16 – Attributs de l'objet "Compact data" 612
Tableau 17 – Attributs de l'objet "Compact data" 612
Tableau 18 – Codage A-XDR des données lues dans l'attribut *buffer* 613
Tableau 19 – Paramètres pour l'accès sélectif aux attributs object_list et access_rights_list 617
Tableau 20 – Paramètres pour l'accès sélectif à l'attribut object_list 621
Tableau 21 – Codage des paramètres d'accès sélectif avec data_index 654
Tableau 22 – Informations relatives aux clés exigées pour établir les clés de protection des données 665
Tableau 23 – Paramètres de protection de l'attribut *protection_parameters_get* 666
Tableau 24 – Paramètres de protection de l'attribut *protection_parameters_set* 667
Tableau 25 – Paramètres de protection de la méthode *get_protected_attributes* 668
Tableau 26 – Paramètres de protection de la méthode *set_protected_attributes* 669
Tableau 27 – Paramètres de protection de la méthode *invoke_protected_method* 670
Tableau 28 – Exemples de valeurs pour NCA et CLT 679
Tableau 29 – Schedule 686
Tableau 30 – Special days table 686
Tableau 31 – IC "Disconnect control" – États et transitions d'état 699
Tableau 32 – Présentation explicite des tableaux de valeurs de seuils 711
Tableau 33 – Présentation explicite d'action_sets 711
Tableau 34 – États du crédit 729
Tableau 35 – Transitions d'état du crédit 730
Tableau 36 – Éléments de l'adresse ADS 754
Tableau 37 – Registre des erreurs fatales 755
Tableau 38 – Mise en correspondance des variables MIB de l'IEC 61334-4-512:2001 avec les attributs/méthodes des IC COSEM 800

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 39 – Adresses MAC dans le profil S-FSK 806
Tableau 40 – Mise en correspondance des attributs PRIME NB OFDM PLC PIB avec les attributs d'IC COSEM 822
Tableau 41 – Mise en correspondance des attributs G3-PLC PIB avec les attributs d'IC COSEM 835
Tableau 42 – Utilisation des classes d'interfaces COSEM pour l'établissement ZigBee® 859
Tableau 43 – Règle C/D 1 875
Tableau 44 – Paramètres pour l'accès sélectif à l'attribut object_list 912
Tableau 45 – Paramètres pour l'accès sélectif aux attributs *object_list* et *access_rights_list* 917
Tableau 46 – Paramètres pour l'accès sélectif à l'attribut object_list 921
Tableau 47 – Paramètres pour l'accès sélectif à l'attribut *object_list* 927
Tableau 48 – Codage des paramètres d'accès sélectif avec data_index 946
Tableau 49 – Éléments de l'adresse ADS 959
Tableau 50 – Mise en correspondance des attributs PIB G3-PLC spécifiés dans l'UIT-T G.9903:2017 avec les attributs d'IC COSEM 985
Tableau 51 – Mise en correspondance des attributs G3-PLC PIB avec les attributs d'IC COSEM 998
Tableau 52 – Utilisation du groupe de valeurs C pour des objets abstraits dans le contexte COSEM 1012
Tableau 53 – Représentation de diverses valeurs par les IC appropriées 1043
Tableau 54 – Algorithmes de mesure – Valeurs énumérées 1046
Tableau 55 – Objets "Threshold", électricité 1049
Tableau 56 – Objets "Register monitor", électricité 1049
Tableau B.1 – Clé de cryptage prédéfinie dans l'esclave et ne pouvant pas être modifiée 1054
Tableau B.2 – Clé de cryptage prédéfinie dans l'esclave et définition d'une nouvelle clé après installation 1054
Tableau B.3 – Clé de cryptage non prédéfinie dans l'esclave, mais pouvant être définie, cas a) 1054
Tableau B.4 – Clé de cryptage non prédéfinie dans l'esclave, mais pouvant être définie, cas b) 1054
Tableau C.1 – En-têtes IPv6 par rapport à l'IC IPv6 1058
Tableau C.2 – Extensions d'en-tête IPv6 facultatives par rapport à l'IC IPv6 1058

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

ÉCHANGE DES DONNÉES DE COMPTAGE DE L'ÉLECTRICITÉ – LA SUITE DLMS®/COSEM –

Partie 6-2: Classes d'interfaces COSEM

AVANT-PROPOS

1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – 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 l'IEC"). 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 l'IEC, participent également aux travaux. L'IEC 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 l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC 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 l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC 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 l'IEC. L'IEC 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 l'IEC, 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 l'IEC, 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 l'IEC ou de toute autre Publication de l'IEC, 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 du présent document de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

L'IEC 62056-6-2 a été établie par le comité d'études 13 de l'IEC: Comptage et pilotage de l'énergie électrique. Il s'agit d'une Norme internationale.

Cette quatrième édition annule et remplace la troisième édition de l'IEC 62056-6-2 parue en 2017. Cette édition constitue une révision technique.

Les modifications techniques majeures par rapport à l'édition précédente sont énumérées à l'Annexe F (informative).

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
13/1891/FDIS	13/1906/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La version française de la norme n'a pas été soumise au vote.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/publications.

Une liste de toutes les parties de la série IEC 62056, publiées sous le titre général *Échange des données de comptage de l'électricité – La suite DLMS®/COSEM*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo "colour inside" qui se trouve sur la page de couverture de ce document 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.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

INTRODUCTION

La présente quatrième édition de l'IEC 62056-6-2 a été établie par le groupe de travail 14 du comité d'études 13 de l'IEC avec la contribution significative de la DLMS® User Association, son partenaire de liaison de type A.

La présente édition est conforme à l'Édition 14 du Livre Bleu de la DLMS® UA. Les principales nouvelles propriétés sont l'IC "Array manager", la version 1 de l'IC "Compact data", la version 1 de l'IC "GSM diagnostic", l'IC "LTE monitoring", l'IC "NTP setup", les IC "HS-PLC setup" et les nouveaux codes OBIS associés.

Modélisation des objets et identification des données

En réponse aux besoins professionnels des acteurs du marché de l'énergie – en général dans un environnement concurrentiel libéralisé – et au souhait de gérer efficacement les ressources naturelles tout en impliquant le consommateur, le compteur est devenu un composant à part entière d'un système intégré de comptage, de contrôle et de facturation. Le compteur n'est plus un simple dispositif d'enregistrement de données et il s'appuie surtout sur des capacités de communication. La facilité d'intégration, l'interopérabilité et la sécurité des données du système sont des exigences importantes.

La spécification COSEM (*Companion Specification for Energy Metering*) relève ces défis en considérant le compteur comme une partie d'un système de mesure et de contrôle complexe. Le compteur doit pouvoir transférer les résultats de mesure des points de comptage vers les processus métier qui les utilisent. Il doit également pouvoir fournir des informations au consommateur et gérer la consommation et finalement la production locale.

Pour y parvenir, la COSEM utilise des techniques de *modélisation d'objets* afin de modéliser toutes les fonctions du compteur, sans formuler d'hypothèses sur les fonctions à prendre en charge, la manière dont ces fonctions sont mises en œuvre et la façon dont les données sont transportées. La spécification formelle des classes d'interfaces COSEM constitue une partie prépondérante de la COSEM.

Pour traiter et gérer les informations, il est indispensable d'identifier de manière unique tous les éléments de données indépendamment du fabricant. La définition du système OBIS (*Object Identification System)* est une autre partie essentielle de la COSEM. Elle repose sur la DIN 43863-3:1997, *Elektrizitätszähler – Teil 3: Tarifgeräte als Zusatzeinrichtung zum Elektrizitätszähler – EDIS – Energie-Daten-Identifikations-System* (disponible en allemand seulement). Le jeu de codes OBIS a été considérablement étendu au fil des années de manière à répondre aux nouveaux besoins.

La COSEM modélise le compteur comme une application *serveur* (voir 4.1.7) utilisée par les applications *clients* qui récupèrent les données du compteur, transmettent des informations de commande au compteur et déclenchent des actions connues dans le compteur via l'accès contrôlé aux objets COSEM. Les *clients* agissent comme des agents pour des tiers, c'est-à-dire les processus métiers des acteurs du marché de l'énergie.

Les classes d'interfaces COSEM normalisées forment une bibliothèque extensible. Les fabricants utilisent les éléments de cette bibliothèque pour concevoir leurs produits qui satisfont à un large éventail d'exigences.

Le serveur offre des moyens de récupération des fonctions prises en charge, c'est-à-dire les objets COSEM instanciés. Les objets peuvent être organisés en *dispositifs logiques* et en *associations d'applications*, et donnent des droits d'accès particuliers aux différents clients.

Le concept de la bibliothèque normalisée de classes d'interfaces fournit aux différents utilisateurs et fabricants un maximum de diversité tout en assurant l'interopérabilité.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

La Commission Électrotechnique Internationale (IEC) attire l'attention sur le fait qu'il est déclaré que la conformité aux dispositions de la présente Norme internationale peut impliquer l'utilisation d'un service de maintenance concernant la pile de protocoles sur laquelle est basée la présente Norme IEC 62056-6-2.

L'IEC ne prend pas position quant à la preuve, la validité et la portée de ce service de maintenance.

Le fournisseur du service de maintenance a donné l'assurance à l'IEC qu'il consent à fournir des services à des demandeurs du monde entier, en des termes et à des conditions raisonnables et non discriminatoires. À cet égard, la déclaration du fournisseur du service de maintenance est enregistrée à l'IEC. Des informations peuvent être obtenues auprès de:

DLMS[1] User Association (Association des utilisateurs de la DLMS)
www.dlms.com

Remerciements

Le document actuel a été établi par le groupe de travail Maintenance de DLMS® UA.

Les paragraphes 4.4.7 et 4.4.9 reposent sur des documents du NIST. Reproduits avec l'aimable autorisation du National Institute of Standards and Technology, Technology Administration, U.S. Department of Commerce. Non protégés par des droits d'auteur aux États-Unis.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

[1] Device Language Message Specification (Spécification des messages de langage d'équipement).

ÉCHANGE DES DONNÉES DE COMPTAGE DE L'ÉLECTRICITÉ – LA SUITE DLMS®/COSEM –

Partie 6-2: Classes d'interfaces COSEM

1 Domaine d'application

La présente partie de l'IEC 62056 spécifie un modèle de compteur tel qu'il est vu à travers son ou ses interfaces de communication. Des blocs génériques de base sont définis à l'aide de méthodes orientées objets, sous forme de classes d'interfaces permettant de modéliser les compteurs ayant des fonctionnalités allant de simples à très complexes.

Les Annexes A à F (informatives) donnent des informations supplémentaires relatives à certaines classes d'interfaces.

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. 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).

EN 13757-2:2004, *Systèmes de communication et de télérelevé de compteurs – Partie 2:Couches physiques et couche de liaison*

EN 13757-3:2004, *Systèmes de communication et de télérelevé de compteurs – Partie 3:Couche d'application spéciale*

NOTE Cette norme est référencée dans la version 0 de la classe d'interfaces "M-Bus client setup".

EN 13757-3:2013, *Systèmes de communication et de télérelevé de compteurs – Partie 3:Couche d'application spéciale*

NOTE Cette norme est référencée dans la version 1 de la classe d'interfaces "M-Bus client setup".

EN 13757-4:2013, *Systèmes de communication et de télérelevé des compteurs – Partie 4:Échange de données des compteurs par radio (lecture de compteurs dans la bande SRD)*

EN 13757-5:2015, *Systèmes de communication pour compteurs – Partie 5:Relais de transmission sans fil M-Bus*

IEC 61334-4-32:1996, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4:Protocoles de communication de données – Section 32:Couche liaison de données – Contrôle de liaison logique (LLC)*

IEC 61334-4-41:1996, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4:Protocoles de communication de données – Section 41:Protocoles d'application – Spécification des messages de ligne de distribution*

IEC 61334-4-511:2000, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4-511:Protocoles de communication de données – Administration de systèmes – Protocole CIASE*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC 61334-4-512:2001, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 4-512:Protocoles de communication de données – Administration de systèmes à l'aide du profil 61334-5-1 – MIB (Base d'Informations d'Administration)*

IEC 61334-5-1:2001, *Automatisation de la distribution à l'aide de systèmes de communication à courants porteurs – Partie 5-1:Profils des couches basses – Profil S-FSK (modulation par saut de fréquences étalées)*

IEC TR 62055-21:2005, *Electricity metering – Payment systems – Part 21:Framework for standardization* (disponible en anglais seulement)

IEC 62056-21:2002, *Équipement de mesure de l'énergie électrique – Échange des données pour la lecture des compteurs, le contrôle des tarifs et de la charge – Partie 21:Échange des données directes en local*

IEC 62056-31:1999, *Comptage de l'électricité – Échange de données pour la lecture des compteurs, le contrôle des tarifs et de la charge – Partie 31:Utilisation des réseaux locaux sur paire torsadée avec signal de porteuse*

NOTE Cette édition est référencée dans la classe d'interfaces "IEC twisted pair (1) setup" (class_id: 24, version: 0).

IEC 62056-3-1:2013, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 3-1:Utilisation des réseaux locaux sur paire torsadée avec signal de porteuse*

NOTE Cette édition est référencée dans la classe d'interfaces "IEC twisted pair (1) setup" (class_id: 24, version: 1).

IEC 62056-3-1:2021, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 3-1:Utilisation des réseaux locaux sur paire torsadée avec signal de porteuse*

IEC 62056-46:2002/A1:2006, *Équipements de mesure de l'énergie électrique – Échange des données pour la lecture des compteurs, le contrôle des tarifs et de la charge – Partie 46:Couche liaison utilisant le protocole HDLC*

IEC 62056-5-3:2023, *Échange des données de comptage de l'électricité – La suite DLMS®/COSEM – Partie 5-3:Couche application DLMS®/COSEM*

IEC 62056-6-1:2023, *Échange des données de comptage de l'électricité – La suite DLMS®/COSEM – Partie 6-1:Système d'identification des objets (OBIS)*

IEC 62056-7-3:2017, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 7-3: Profils de communication M-Bus filaire et sans fil pour les réseaux locaux et les réseaux de voisinage*

IEC 62056-8-3:2013, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 8-3:Profil de communication pour réseaux de voisinage CPL S-FSK*

IEC 62056-8-4:2018, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 8-4:Profils de communication pour réseaux de voisinage OFDM PLC PRIME à bande étroite*

IEC 62056-8-6:2017, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 8-6:Profil CPL ISO/IEC 12139-1 à grande vitesse pour les réseaux de voisinage*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC 62056-8-8:2020, *Échange des données de comptage de l'électricité – La suite DLMS/COSEM – Partie 8-8:Profil de communication pour réseaux de la série ISO/IEC 14908*

IEEE 802.15.4:2006, *Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)* (disponible en anglais seulement)

NOTE Cette norme est également disponible en tant qu'ISO/IEC/IEEE 8802-15-4:2010.

IETF STD 51, *The Point-to-Point Protocol (PPP)*, 1994 (voir aussi: RFC 1661, RFC 1662)

ISO/IEC 8802-2:1998, *Technologies de l'information – Télécommunications et échange d'information entre systèmes – Réseaux locaux et métropolitains – Exigences spécifiques – Partie 2:Contrôle de liaison logique*

ISO/IEC 12139-1:2009, *Technologies de l'information – Télécommunications et échange d'information entre systèmes – Courants porteurs en ligne (PLC) – Contrôle d'accès au support (MAC) et couche physique (PHY) par PLC à grande vitesse – Partie 1:Exigences générales*

ISO/IEC 14908-1:2012, *Technologies de l'information – Protocole de réseau de contrôle – Partie 1:Pile de protocoles*

ISO/IEC 60559:2020, *Technologies de l'information – Systèmes de microprocesseurs – Arithmétique flottante*

ISO 4217, *Codes pour la représentation des monnaies*

UIT-T E.212 (05.2008), *Série E:Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains – Exploitation des relations internationales – Service mobile maritime et service mobile terrestre public – Plan d'identification international pour les réseaux publics et les abonnements*

UIT-T G.9903 Amd. 1:2013, *Série G:Systèmes et supports de transmission, systèmes et réseaux numériques – Réseaux d'accès – Réseaux intérieurs – Émetteurs-récepteurs OFDM à bande étroite utilisant les courants porteurs en ligne – G3-PLC*

NOTE Cette recommandation est référencée dans la version 0 des classes "G3-PLC setup".

UIT-T G.9903:2014, *Série G:Systèmes et supports de transmission, systèmes et réseaux numériques – Réseaux d'accès – Réseaux intérieurs – Émetteurs-récepteurs OFDM à bande étroite utilisant les courants porteurs en ligne – G3-PLC*

NOTE Cette recommandation est référencée dans la version 1 des classes "G3-PLC setup".

UIT-T G.9903 Amd. 1:2017, *Series G:Systèmes et supports de transmission, systèmes et réseaux numériques – Réseaux d'accès – Réseaux intérieurs – Émetteurs-récepteurs OFDM à bande étroite utilisant les courants porteurs en ligne – G3-PLC*

UIT-T G.9904:2012, *Série G:Systèmes et supports de transmission, systèmes et réseaux numériques – Réseaux d'accès – Réseaux intérieurs – Émetteurs-récepteurs OFDM à bande étroite utilisant les courants porteurs en ligne – PRIME*

ETSI GSM 05.08:1996, *Digital cellular telecommunications system (Phase 2+); Radio subsystem link control* (disponible en anglais seulement)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3 Termes, définitions et abréviations

Pour les besoins du présent document, les termes et définitions suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

3.1 Termes et définitions liés au processus de transfert d'image (voir 4.4.6)

3.1.1
Image
données binaires de taille spécifiée

Note 1 à l'article: Une Image peut être perçue comme un conteneur. Elle peut être composée d'un ou plusieurs éléments (image_to_activate) qui sont transférés, vérifiés et activés ensemble.

3.1.2
ImageSize
taille de l'ensemble de l'Image à transférer

Note 1 à l'article: ImageSize est exprimé en octets.

3.1.3
ImageBlock
partie de l'Image de la taille ImageBlockSize

Note 1 à l'article: L'Image est transférée dans des ImageBlocks. Chaque bloc est identifié par son ImageBlockNumber.

3.1.4
ImageBlockSize
taille de l'ImageBlock en octets

3.1.5
ImageBlockNumber
identifiant d'un ImageBlock. Les ImageBlocks sont numérotés séquentiellement, à partir de 0

La signification des définitions ci-dessus est représentée à la Figure 1.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

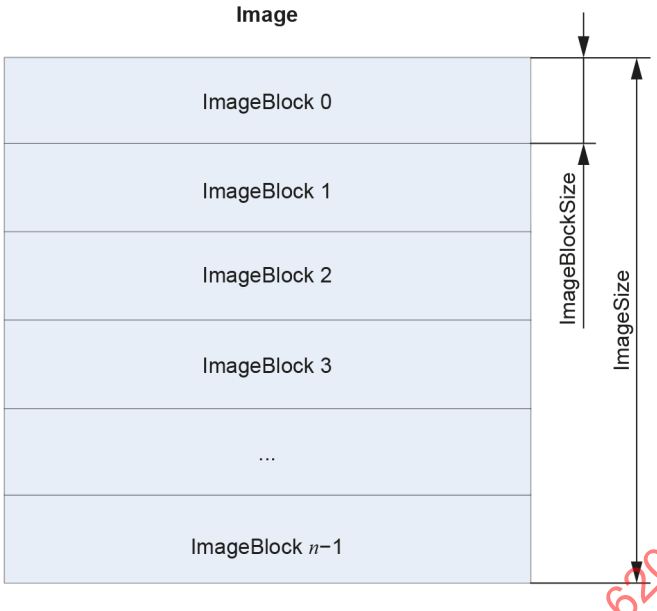


Figure 1 – Signification des définitions concernant l'Image

3.2 Termes et définitions liés aux classes "S-FSK PLC setup" (voir 4.10)

3.2.1
initiateur
élément utilisateur d'une entité d'application de l'administration de systèmes (SMAE) client

Note 1 à l'article: L'initiateur utilise les ASE de CIASE et de xDLMS et est identifié par son titre de système.

[SOURCE: IEC 61334-4-511:2000, 3.8.1]

3.2.2
initiateur actif
initiateur qui émet ou vient d'émettre une demande Register (recensement) CIASE lorsque le serveur est en état non configuré

[SOURCE: IEC 61334-4-511:2000, 3.9.1]

3.2.3
nouveau système
système serveur en état non configuré: son adresse MAC correspond à l'adresse NEW

[SOURCE: IEC 61334-4-511:2000, 3.9.3]

3.2.4
titre de nouveau système
titre de système d'un nouveau système

Note 1 à l'article: Il s'agit du titre de système dans le cas d'un système qui est dans le nouvel état.

[SOURCE: IEC 61334-4-511:2000, 3.9.4]

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.2.5
système enregistré
système serveur ayant une adresse MAC valide individuelle (donc différente de l'adresse NEW, voir IEC 61334-5-1:2001: Contrôle d'accès au support)

[SOURCE: IEC 61334-4-511:2000, 3.9.5]

3.2.6
système de notification
système serveur qui émet une DiscoverReport (notification de recherche)

[SOURCE: IEC 61334-4-511:2000, 3.9.6, modifié pour corriger une erreur dans l'IEC 61334-4-511]

3.2.7
sous-intervalle
temps nécessaire pour émettre deux octets par la couche physique

Note 1 à l'article: Les intervalles de temps sont divisés en sous-intervalles dans le mode RepeaterCall de la couche physique.

3.2.8
intervalle de temps
temps nécessaire pour émettre une trame physique

Note 1 à l'article: Comme spécifié dans l'IEC 61334-5-1:2001, 3.3.1, une trame physique comprend un préambule de 2 octets, un délimiteur de sous-trame de début de 2 octets, une PSDU de 38 octets et une pause de 3 octets.

3.3 Termes et définitions liés aux classes d'interfaces "PRIME NB OFDM PLC setup" (voir 4.12)

Définitions liées à la couche physique

3.3.1
nœud de base
nœud principal qui commande et gère les ressources d'un sous-réseau

[SOURCE: UIT-T G.9904:2012, 3.2.1]

3.3.2
emplacement d'étiquette
emplacement de l'étiquette PDU dans la trame

[SOURCE: UIT-T G.9904:2012, 3.2.2]

3.3.3
nœud
élément d'un sous-réseau, qui est en mesure d'émettre et de recevoir à partir d'autres éléments du sous-réseau

[SOURCE: UIT-T G.9904:2012, 3.2.9]

3.3.4
enregistrement
processus par lequel un nœud de service est accepté comme membre du sous-réseau, auquel un LNID est attribué

[SOURCE: UIT-T G.9904:2012, 3.2.12]

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.3.5
nœud de service
nœud d'un sous-réseau, qui n'est pas un nœud de base

[SOURCE: UIT-T G.9904:2012, 3.2.13]

3.3.6
sous-réseau
ensemble d'éléments qui peuvent communiquer en se conformant à la présente spécification et partager un seul nœud de base

[SOURCE: UIT-T G.9904:2012, 3.2.15]

Définitions liées à la couche MAC

3.3.7
état déconnecté
<d'un nœud de service> état fonctionnel initial de tous les nœuds de service. Lorsqu'il est déconnecté, un nœud de service ne peut ni communiquer de données ni commuter les données d'autres nœuds. Sa fonction principale consiste à rechercher un sous-réseau à sa portée et de tenter de s'y enregistrer

[SOURCE: UIT-T G.9904:2012, 8.1]

3.3.8
état final
<d'un nœud de service> lorsqu'il est dans cet état fonctionnel, un nœud de service peut établir des connexions et communiquer des données, mais il ne peut pas commuter des données d'autres nœuds

[SOURCE: UIT-T G.9904:2012 8.1]

3.3.9
état de commutation
<d'un nœud de service> lorsqu'il est dans cet état fonctionnel, un nœud de service peut réaliser toutes les fonctions finales. De plus, il peut transférer des données entre d'autres nœuds du même sous-réseau. Il s'agit d'un point de branchement sur la structure de l'arborescence

[SOURCE: UIT-T G.9904:2012, 8.1]

3.3.10
promotion
processus par lequel un nœud de service est qualifié pour commuter (répéter, transférer) le trafic de données provenant d'autres nœuds et faire office de point de branchement sur la structure de l'arborescence du sous-réseau. Une promotion réussie représente la transition entre l'état final et l'état de commutation. Lorsqu'un nœud de service est à l'état déconnecté, il ne peut pas passer directement à l'état de commutation

[SOURCE: UIT-T G.9904:2012, 8.1]

3.3.11
rétrogradation
processus par lequel un nœud de service cesse d'être un point de branchement sur la structure de l'arborescence de sous-réseau. Une rétrogradation réussie représente la transition entre l'état de commutation et l'état final

[SOURCE: UIT-T G.9904:2012, 8.1]

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.4 Termes et définitions liés aux classes d'interfaces "ISO/IEC 14908 setup" (voir 4.18)

3.4.1
domaine
réseau logique qui est une unité d'adressage

Note 1 à l'article: Les adresses des sous-réseaux (voir ci-dessous) et des nœuds sont attribuées par l'administrateur responsable du domaine, et elles n'ont de signification que dans le contexte de ce domaine.

Note 2 à l'article: Tous les nœuds doivent faire partie du même domaine pour être mutuellement adressables.

3.4.2
nœud
concept lié à un nœud physique qui représente le plus haut degré de résolvabilité des adresses sur un réseau

Note 1 à l'article: Un nœud est identifié (adressé) au sein d'un sous-réseau par son identifiant de nœud (logique). Un nœud physique peut appartenir à plusieurs sous-réseaux. Dans ce cas, un numéro de nœud (logique) lui est attribué pour chaque sous-réseau auquel il appartient. Un nœud physique peut appartenir au plus à deux sous-réseaux; ces sous-réseaux doivent être dans des domaines différents. Un nœud peut également être identifié (sans ambiguïté) au sein d'un réseau par son Unique_Node_ID.

3.4.3
sous-réseau
ensemble de nœuds accessibles via le même protocole de couche liaison

3.4.4
transaction
séquence de messages corrélés entre eux

3.5 Termes et définitions liés à ZigBee® (voir 4.15)

NOTE Les termes suivis d'un astérisque (*) sont issus de la spécification ZigBee®.

3.5.1
CAD
dispositif de passerelle ZigBee® qui agit comme un IHD à l'intérieur du réseau ZigBee®, mais qui comporte une connexion supplémentaire à un autre réseau (c'est-à-dire WiFi)

3.5.2
IHD
dispositif comportant un écran pour l'affichage des informations énergétiques au consommateur

3.5.3
code d'installation
clé de liaison préconfigurée (PCLK) hachée (via MMO) fournie à un tiers de confiance par l'intermédiaire des communications hors bande. Un nouveau dispositif qui souhaite intégrer le réseau doit envoyer ce code d'installation au tiers de confiance, ce qui permet à ce dernier d'exécuter le processus d'accès, en utilisant ce code d'installation comme partie intégrante des informations de sécurité

3.5.4
clé de liaison *
clé partagée exclusivement entre deux, et seulement deux, entités de couche application homologues au sein d'un réseau PAN

3.5.5
adresse MAC/adresse IEEE
termes utilisés en synonymes pour représenter le code EUI-64 attribué à ZigBee® Radio

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.5.6
ZigBee®
spécification d'une suite de protocoles de communication de haut niveau utilisée pour créer des réseaux personnels conçus à partir de petites radios numériques basse puissance. ZigBee® repose sur une norme IEEE 802.15. Malgré leur faible puissance, les dispositifs ZigBee® transmettent souvent des données sur de plus longues distances, en passant par des dispositifs intermédiaires pour atteindre des dispositifs plus éloignés, en créant ainsi un réseau maillé

3.5.7
client ZigBee®
s'apparente au rôle du client DLMS®/COSEM

Note 1 à l'article: Pour mieux comprendre l'interaction entre le client et le serveur, il convient de lire la spécification ZigBee® PRO.

3.5.8
coordinateur ZigBee® *
coordinateur de réseau PAN IEEE 802.15.4-2006 qui est le principal contrôleur d'un réseau IEEE 802.15.4-2006 chargé de la formation du réseau. Le coordinateur de réseau PAN doit être un dispositif à fonction intégrale (FFD)

3.5.9
cluster ZigBee®
ensemble de types de messages associés à une certaine fonction de dispositif (par exemple, comptage ou commande de ballast)

3.5.10
miroir ZigBee®
dispositif qui renvoie par écho les données publiées par un dispositif ZigBee® alimenté par batterie, permettant aux autres acteurs du réseau d'obtenir les données lorsque le dispositif alimenté par batterie est indisponible pour des raisons d'économie d'énergie

3.5.11
ZigBee® PRO
autre nom du protocole ZigBee® 2007. ZigBee® 2007, qui est à présent la nouvelle édition de pile, contient deux profils de pile: le profil de pile 1 (tout simplement appelé ZigBee®), pour les utilisations domestiques et commerciales légères, et le profil de pile 2 (appelé ZigBee® PRO). ZigBee® PRO offre plus de fonctionnalités, comme la multidiffusion, l'acheminement à origines multiples et destination unique et la haute sécurité avec l'échange de clés symétriques (SKKE), ZigBee® (profil de pile 1) offrant une plus petite empreinte dans la mémoire RAM et Flash. Les deux protocoles offrent un réseau maillé complet et fonctionnent avec tous les profils d'application ZigBee®

3.5.12
routeur ZigBee® *
FFD IEEE 802.15.4-2006 participant à un réseau ZigBee®, qui n'est pas le coordinateur ZigBee®, mais qui peut agir comme un coordinateur IEEE 802.15.4-2006 dans son espace de fonctionnement personnel, capable d'acheminer des messages entre des dispositifs et de prendre en charge des associations

3.5.13
serveur ZigBee®
s'apparente au rôle du serveur DLMS®/COSEM

Note 1 à l'article: Pour mieux comprendre l'interaction entre le client et le serveur, il convient de lire la spécification ZigBee® PRO.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.5.14
tiers de confiance ZigBee® *
dispositif digne de confiance au sein d'un réseau ZigBee® pour distribuer les clés aux fins de la gestion des configurations du réseau et des applications de bout en bout

3.6 Termes et définitions liés aux classes d'interfaces de comptage à paiement (voir 4.6)

3.6.1
compte
déclaration des crédits et des charges d'un individu dans le cadre d'une relation contractuelle entre ledit individu et une autre partie (un fournisseur de services en l'occurrence)

3.6.2
disponible
(crédit) valeur totale qui peut être décrémentée des charges sans autre action

3.6.3
charge
représentation du passif financier d'un compte

Note 1 à l'article: Dans la présente spécification, les charges sont modélisées sous forme d'objets "Charge" qui définissent le montant dû, le mécanisme de collecte, la périodicité de la collecte, le montant de la collecte, ainsi que d'autres variables pertinentes.

Note 2 à l'article: Il peut y avoir un ou plusieurs versements, leur montant pouvant être déterminé explicitement ou en termes de taux de paiement par unité de temps ou de consommation.

Note 3 à l'article: Les charges peuvent également être prélevées en montant fixe à la vente.

3.6.4
mode de crédit
mode de fonctionnement d'un compteur dans un système de paiement qui n'exige pas le paiement à l'avance de la consommation

3.6.5
collecter
percevoir le paiement d'un acompte d'une charge, en tenant compte du montant déterminé par l'attribut *unit_charge_active* de l'objet "Charge"

3.6.6
marchandise
produit livré à un consommateur à un point de service de ses locaux dans le cadre d'un contrat d'approvisionnement (électricité, gaz, eau et chauffage, par exemple)

3.6.7
activé
Lorsqu'il est utilisé dans le contexte des types "Credit" ou "Charge", signifie que le type "Credit" ou "Charge" apparaît respectivement dans la*credit_reference_list* ou la *charge_reference_list* de l'objet "Account"

3.6.8
crédit d'urgence
montant d'un crédit saisi dans un système de comptage à paiement fonctionnant en mode de prépaiement, représentant un prêt à court terme consenti au consommateur

Note 1 à l'article: Il s'agit d'une fonctionnalité de certains systèmes de comptage à paiement dans lesquels le consommateur peut obtenir un prêt à court terme limité, souvent consenti en local par l'unité de prépaiement elle-même. Le terme "urgence" indique un besoin urgent plutôt qu'un sinistre.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

3.6.9
progiciel de gestion intégré (ERP)
application d'arrière-guichet
système informatique qui réalise les processus métier d'une organisation (un fournisseur d'énergie, par exemple), par opposition au système de communication. (Voir aussi "système de tête de réseau")

3.6.10
crédit à l'amiable
période avec un début et une fin configurables, au cours de laquelle le compteur ne déconnecte pas l'alimentation quel que soit l'état d'*available_credit*. Également appelé "période de non-déconnexion"

Note 1 à l'article: Cette fonction est utilisée lorsqu'il n'est pas envisageable d'obtenir le crédit nécessaire (la nuit ou dans le cas de personnes âgées fragiles, par exemple).

3.6.11
système de tête de réseau
HES
système informatique, connecté par un réseau de communication à une population de dispositifs intelligents, dont le travail consiste à piloter et coordonner les flux d'informations entre ces dispositifs, en général pour le compte d'un ERP ("application d'arrière-guichet") séparé

3.6.12
réseau local domestique
HAN
réseau de communication construit dans le principal but de connecter les dispositifs dans un seul local

3.6.13
en cours d'utilisation
état d'un objet "Credit" qui, au moment de la requête, dispose d'un *current_credit_amount* positif, ce crédit étant consommé par certaines charges actives représentées par les objets "Charge"

Note 1 à l'article: Lorsque *current_credit_amount* devient nul, le crédit est épuisé.

3.6.14
limitation de la charge
mode de fonctionnement de certains systèmes de comptage à paiement (pas nécessairement en mode de prépaiement) dans lequel le consommateur peut recevoir une alimentation à condition de ne pas dépasser un niveau de demande configuré

Note 1 à l'article: Il s'agit implicitement de gérer les finances du consommateur: si la demande est limitée au profit du système de génération ou de distribution, le terme "gestion de la charge" est plus souvent utilisé.

3.6.15
communications locales
mécanisme de communication avec le compteur sur certains supports, au voisinage dudit compteur (sur un réseau local domestique ou un port optique, par exemple)

3.6.16
entrée manuelle
entrée d'un jeton dans l'installation de comptage à paiement au moyen d'un processus manuel

3.6.17
mode de paiement géré
spécialisation du mode de crédit qui permet d'activer un compte, un crédit, voire des charges, dans un compteur lorsque le paiement du service est reçu par le système après que ce service a été consommé

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Note 1 à l'article: En règle générale, en mode de paiement géré, les jetons ne sont pas utilisés. Toutefois, le crédit est ajusté à l'aide des méthodes définies dans l'objet "Credit".

Note 2 à l'article: Le compteur tolère un nombre admissible de dettes avant d'être crédité de la part du client parallèlement à un paiement en espèce de la part du système.

Note 3 à l'article: Dans cet exemple, le terme "espèce" est un terme générique pour indiquer un paiement réel adressé au système, et qui pourrait être exécuté en monnaie ayant cours légal, par transfert électronique automatisé, etc.

3.6.18
installation du comptage à paiement
ensemble d'équipements de comptage à paiement installés et prêts à l'emploi dans les locaux d'un consommateur

Note 1 à l'article: Cette installation comprend le montage des équipements et, dans le cas d'une installation à plusieurs dispositifs, le raccordement de chaque équipement. Elle inclut également le raccordement du réseau d'alimentation à chaque interface d'alimentation, le raccordement de l'interface de charge du consommateur et la mise en service de l'équipement à l'état opérationnel comme une installation de comptage à paiement.

3.6.19
mode de prépaiement
mode de fonctionnement d'un compteur dans un système de paiement, par lequel le consommateur paye un service avant la consommation effective

3.6.20
post-paiement
méthode de fonctionnement d'un système de paiement par lequel un consommateur peut consommer le service avant de le payer

Note 1 à l'article: Ce terme peut être remplacé par le terme "mode de crédit" lorsqu'il est utilisé dans le contexte des modes opérationnels.

Note 2 à l'article: Ce terme est généralement utilisé conjointement avec une description du système, alors que le mode de crédit est utilisé lorsqu'il est fait référence au mode opérationnel d'un compteur ou d'un compte.

3.6.21
communications distantes
transport d'un jeton ou d'un autre message entre un client et un serveur qui exécute un processus d'application de comptage à paiement via une certaine forme de réseau sans fil et de réseau d'accès. Il peut s'agir d'une connexion point à point, à maillage radio, à fibre optique, etc., et qui passer par plusieurs dispositifs et plusieurs protocoles avant d'atteindre le compteur

3.6.22
remboursable
types de crédits (un crédit d'urgence, par exemple) dont un montant ajouté au *current_credit_amount* d'un objet "Credit" doit être remboursé avant de pouvoir de nouveau sélectionner le crédit

3.6.23
crédit réservé
montant du crédit mis en réserve dans le compte d'un compteur à paiement, en vue d'une utilisation ultérieure, au choix du consommateur

Note 1 à l'article: Le mécanisme de réservation de ce crédit peut faire l'objet d'un accord entre le fournisseur du système et le consommateur. Par exemple, une partie des jetons peut être ajoutée au crédit de réserve ou le fournisseur peut donner une allocation mensuelle au consommateur, mais ces dispositions sont spécifiques au projet.

3.6.24
sélectionnable
état spécifique d'un objet "Credit" dans lequel la confirmation immédiate du consommateur est nécessaire avant de pouvoir l'utiliser

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Note 1 à l'article: Par exemple, le crédit d'urgence est, par nature, un prêt à court terme et il convient donc de ne le déployer qu'avec le consentement du consommateur. Ce terme fait respectivement référence à la nécessité de conclure un accord et au fait d'avoir obtenu un accord. Seul un crédit (1) sélectionnable peut être (2) sélectionné/appelé. Tous les crédits ne sont pas sélectionnables par un déclencheur externe, car, dans la plupart des cas, l'application du compteur exécute automatiquement cette action.

3.6.25
sélectionné/appelé
état spécifique d'un objet "Credit" dans lequel la valeur de *current_credit_amount* est incluse dans le calcul de *available_credit* de l'objet "Account" associé

Note 1 à l'article: Il s'agit de l'état d'un crédit avant qu'il ne devienne en cours d'utilisation, qui est pris en compte dans l'attribut available_credit de l'objet "Account", mais qui n'est pas encore consommé par une charge (un crédit à priorité plus élevée étant en cours d'utilisation).

3.6.26
service
fourniture d'une marchandise (de l'eau, de l'électricité, du gaz ou du chauffage, par exemple)

3.6.27
crédit social
crédit accordé franco de paiement (pour lutter contre la pauvreté, par exemple)

Note 1 à l'article: En règle générale, ce crédit est accordé à périodes fixes (tous les mois, par exemple) en montants limités. Ce type de crédit particulier peut également être basé sur la consommation, le consommateur devant maintenir sa consommation sous un seuil limite afin d'utiliser le crédit social. Cela peut être contrôlé en déconnectant le consommateur si la limite est dépassée.

Note 2 à l'article: Les crédits sociaux sont modélisés sous forme d'objets "Credit" du type emergency_credit, time_based_credit, consumption_based_credit.

3.6.28
dette temporaire
passif provisoire contracté auprès du compteur et qui augmente lorsque les *charge*s sont collectées à un moment où tous les crédits sont épuisés

Note 1 à l'article: Le montant de cette dette temporaire s'accumule dans amount_to_clear de l'objet "Account".

3.6.29
jeton
module de données autonome lié à l'achat du crédit ou à d'autres fonctions du système, et intégré dans un support de jeton (défini ci-après). Le jeton assure la liaison entre la source et la destination de la transaction. Son contenu peut être de l'argent, de l'énergie, du temps, etc., en accord avec la devise déclarée dans le compteur

Note 1 à l'article: L'IEC TR 62055-21:2005 définit ce terme comme suit: "<définition relative à l'équipement>[sic] contenu informatif incluant une instruction émise sur un support de jeton par un système de vente ou de gestion capable de le transférer et d'être accepté par un compteur à paiement spécifique ou par un groupe de compteurs avec une sécurité appropriée".

3.6.30
support de jeton
moyen de transfert d'un jeton entre un élément du système et un autre, souvent dans un format "physique" matériel ou "virtuel" électronique

Note 1 à l'article: Dans un sens général, le jeton fait référence à l'instruction et aux informations transférées, alors que le support de jeton fait référence au dispositif physique utilisé pour transporter l'instruction et les informations ou au support de communication dans le cas d'un support de jeton virtuel.

3.6.31
interface du support de jeton
interface entre le support de jeton et l'installation de comptage à paiement

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Note 1 à l'article: Par exemple, il peut s'agir d'un pavé numérique pour les jetons numériques, d'un utilisateur de support de jeton physique ou d'une connexion de communication à une machine locale ou distante pour une interface de support de jeton virtuel.

Note 2 à l'article: L'interface de support de jeton peut également être utilisée pour transmettre des informations supplémentaires vers/depuis le compteur à paiement, pour les besoins de la gestion du système de paiement.

3.6.32
rechargement
jeton de crédit
crédit contracté par le consommateur et pouvant être délivré sous la forme d'un jeton (et par d'autres moyens) dans un support de jeton physique ou virtuel

3.6.33
communications par dispositif transitoire
transport d'un jeton au sein d'une installation de comptage à paiement par un mécanisme de communication électronique impliquant un dispositif transitoire. Il peut s'agir d'une radio locale, d'une connexion galvanique, d'une connexion optique, etc., provenant de différents dispositifs (terminal portable, téléphone mobile, par exemple)

Note 1 à l'article: Les dispositifs d'affichage à domicile ne sont pas classés dans les dispositifs transitoires, bien qu'ils puissent fonctionner de manière transitoire du point de vue du réseau. Les dispositifs transitoires doivent être considérés comme des dispositifs qui accèdent au réseau pendant une courte durée et à de très rares occasions. Les dispositifs d'affichage à domicile sont considérés comme étant sur le réseau pendant de longues périodes et ne le quittant que sur de courtes périodes comparées à la durée de vie du réseau.

3.6.34
vente
opération ou transaction donnant lieu à un crédit disponible mis sur un compteur à paiement et à augmenter par l'utilisation d'un jeton de crédit

Note 1 à l'article: La vente est souvent liée à une transaction réalisée avec un système de vente installé à un point de vente, permettant de créer un jeton qui peut être transporté au moyen d'un support de jeton physique ou virtuel.

3.7 Termes et définitions liés à la classe d'interfaces Arbitrator (voir 0)

3.7.1
action
opération qui peut être demandée en local ou à distance par le serveur

3.7.2
acteur
entité qui demande une action

Note 1 à l'article: Il peut s'agir d'un processus d'application local ou d'un client.

3.7.3
arbitre
fonction modélisée dans COSEM qui peut déterminer, selon des règles préconfigurées, l'action à réaliser lorsque plusieurs acteurs demandent des actions potentiellement conflictuelles pour contrôler la même source

3.8 Abréviations

Abréviation	Signification
3GPP	3rd Generation Partnership Project
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Network
AA	Application Association (Association d'applications)
AARE	A-Associate Response (Réponse d'association d'applications) – une APDU de l'ACSE
AARQ	A-Associate Request (Demande d'association d'applications) – une APDU de l'ACSE
ACSE	Association Control Service Element (Élément de service de contrôle d'association)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abréviation	Signification
ADP	Primary Station Address (Adresse de la station primaire)
ADS	Secondary Station Address (Adresse de la station secondaire)
AGC	Automatic Gain Control (Commande automatique de gain)
AL	Application Layer (Couche application)
AP	Application Process (Processus d'application)
APDU	Application Protocol Data Unit (Unité de données du protocole d'application)
APS	Application Support Sublayer (Sous-couche de support d'application) (terme ZigBee®)
ARFCN	Absolute Radio-Frequency Channel Number (Numéro de canal radiofréquence absolu)
ASE	Application Service Element (Élément de service d'application)
A-XDR	Adapted Extended Data Representation (Représentation de données étendues adaptée) (IEC 61334-6)
base_name	Attribut short_name correspondant au premier attribut ("logical_name") d'un objet COSEM
BCD	Binary Coded Decimal (Décimal codé binaire)
BER	Bit Error Rate (Taux d'erreurs sur les bits)
CBCP	CallBack Control Protocol (PPP) (Protocole de contrôle du rappel)
CC	Current Credit (Crédit courant) (profil S-FSK PLC)
CDMA	Code Division Multiple Access (Accès multiple par répartition en code)
CENELEC	European Committee for Electrotechnical Standardization (Comité européen de normalisation électrotechnique)
CHAP	Challenge Handshake Authentication Protocol (Protocole d'authentification par défi-réponse)
CIASE	Configuration Initiation Application Service Element (Élément de service d'application d'initiation de configuration) (profil S-FSK PLC)
class_id	Interface class identification code (Code d'identification de classe d'interfaces)
CLI	Calling Line Identity (Identité de la ligne appelante)
COSEM	Companion Specification for Energy Metering (Spécification d'accompagnement pour le comptage de l'énergie)
Objet COSEM	An instance of a COSEM interface class (Instance d'une classe d'interfaces COSEM)
CPAS	Common Part Adaptation Sublayer (Sous-couche d'adaptation de partie commune)
CRC	Cyclic Redundancy Check (Contrôle de redondance cyclique)
CSD	Circuit Switched Data (Transmission de données à commutation de circuits)
CSAP	Client Service Access Point (Point d'accès au service client)
CSMA	Carrier Sense Multiple Access (Accès multiple avec détection de porteuse)
CtoS	Client to Server challenge (Défi du client au serveur)
CU	Currently Unused (Actuellement non utilisé)
DC	Delta Credit (Crédit delta) (profil S-FSK PLC)
DevAddr	(en rapport avec LoRaWAN) Identifiant non unique de 32 bits attribué à un dispositif final de manière statique ou dynamique après une procédure d'accès (en fonction du mode d'activation)
DHCP	Dynamic Host Configuration Protocol (Protocole de configuration dynamique des hôtes)
DIB	Data Information Block (Bloc d'informations de données) (M-Bus)
DIF	Data Information Field (Champ d'informations de données) (M-Bus)
DL	Data Link (Liaison de données)
DLMS®	Device Language Message Specification (Spécification des messages de langage d'équipement)
DLMS® UA	DLMS User Association (Association des utilisateurs de la DLMS)
DLPDU	Data Link Protocol Data Unit (Unité de données de protocole de liaison de données)
DNS	Domain Name Server (Serveur de noms de domaine)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abréviation	Signification
DSCP	Differentiated Services Code Point (Code d'accès aux services différenciés)
DSSID	Direct Switch ID (ID de commutation directe)
EAP	Extensible Authentication Protocol (Protocole d'authentification extensible)
eARFCN	Enhanced Absolute Radio-Frequency Channel Number (Numéro de canal radiofréquence absolu amélioré)
EDGE	Enhanced Data rates for GSM Evolution (GSM à débit amélioré)
EMC	Emergency Credit (Crédit d'urgence) (lié au comptage à paiement)
ERP	Enterprise Resource Planning (Progiciel de gestion intégré)
EUI-48	48-bit Extended Unique Identifier (Identifiant unique étendu à 48 bits)
EUI-64	64-bit Extended Unique Identifier (Identifiant unique étendu à 64 bits)
E-UTRA	Evolved UMTS Terrestrial Radio Acces (Accès radio terrestre UMTS évolué)
FCC	Federal Communications Commission (Commission fédérale des communications)
FFD	Full-Function Device (Dispositif à fonction intégrale)
FIFO	First-In-First-Out (Premier entré, premier sorti)
F/R	Fragmentation and Reassembly (Fragmentation et Réassemblage)
FTP	File Transfer Protocol (Protocole de transfert de fichiers)
GCM	Galois/Counter Mode (Mode Galois/Compteur) – algorithme de cryptage authentifié avec données associées
GMT	Greenwich Mean Time (Temps moyen de Greenwich). Remplacé par le Temps universel coordonné (UTC).
GPRS	General Packet Radio Service (Service général de radiocommunication par paquets)
GPS	Global Positioning System (Système mondial de localisation)
GSM	Global System for Mobile Communications (Système mondial de communications mobiles)
HAN	Home Area Network (Réseau local domestique)
HART	Highway Addressable Remote Transducer (Transducteur distant adressable sur autoroute) (voir http://www.hartcomm.org/) (en lien avec la classe d'interfaces Sensor manager)
HDLC	High-level Data Link Control (Commande de liaison de données à haut niveau)
HES	Head End System (Système de tête de réseau)
HHT	Hand Held Terminal (Terminal portable)
HLS	High Level Security Authentication (Authentification à niveau de sécurité élevé)
HSDPA	High-Speed Downlink Packet Access (Accès par paquets en liaison descendante haut débit)
HS-PLC	High-Speed Power Line Carrier (Courant porteur à grande vitesse)
IANA	Internet Assigned Numbers Authority (Autorité chargée de la gestion de l'adressage sur Internet)
IB	Information Base (Base d'informations)
IC	Interface Class (Classe d'interfaces) (COSEM)
IC	Initial Credit (Crédit initial) (profil S-FSK PLC)
IEC	International Electrotechnical Commission (Commission électrotechnique internationale)
IEEE	Institute of Electrical and Electronics Engineers (Institut des ingénieurs électriciens et électroniciens)
IETF	Internet Engineering Task Force (Groupe de travail d'ingénierie Internet)
IPCP	Internet Protocol Control Protocol (Protocole de commande du Protocole Internet)
IPv4	Internet Protocol version 4 (Protocole Internet version 4)
IPv6	Internet Protocol version 6 (Protocole Internet version 6)
ISO	International Organization for Standardization (Organisation internationale de normalisation)
FAI	Internet Service Provider (Fournisseur d'Accès Internet)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abréviation	Signification
IT	Information Technology (Technologies de l'information)
UIT-T	International Telecommunication Union – Telecommunication (Union Internationale des Télécommunications – Télécommunications)
KEK	Key Encryption Key (Clé de cryptage de clé)
LA	Local Area (Zone locale)
LAC	Local Area Code (Code de zone locale)
LAN	Local Area Network (Réseau local)
LBD	(6)LoWPAN Bootstrapping Device (Dispositif d'amorçage (6)LoWPAN)
LCID	Local Connection Identifier (Identifiant de connexion local)
LCP	Link Control Protocol (Protocole de commande de liaison)
LDN	Logical Device Name (Nom de dispositif logique)
LLC	Logical Link Control (Commande de liaison logique) – sous-couche
LLS	Low Level Security (Niveau de sécurité faible)
LN	Logical Name (Nom logique)
LNID	Local Node Identifier (Identifiant de nœud local)
LOADng	6LoWPAN Ad Hoc On-Demand Distance Vector Routing Next Generation (Protocole de routage à vecteur de distance) (LOADng)
LQI	Link Quality Indicator (Indicateur de qualité de la liaison) (terme ZigBee®)
LSB	Least Significant Bit (Bit de poids faible)
LSID	Local Switch Identifier (Identifiant de commutation local)
LTE	Long Term Evolution (Evolution à long terme) (Communication sans fil)
O	Mandatory (Obligatoire)
M2M	Machine to Machine (Machine à machine)
MAC	Medium Access Control (Contrôle d'accès au support)
M-Bus	Meter Bus (Bus de compteur)
MCC	Mobile Country Code (Code de pays mobile)
MD5	Message Digest Algorithm 5 (Algorithme de condensé de message 5)
MIB	Management Information Base (Base d'informations de gestion) (profil S-FSK PLC)
MID	Measuring Instruments Directive (Directive 2004/22/CE du Parlement européen et du Conseil relative aux instruments de mesure)
MMO	Hachage Matyas-Meyer-Oseas (terme ZigBee®)
MNC	Mobile Network Code (Code de réseau mobile)
MPAN	Meter Point Access Number (terme britannique) – Référence de l'emplacement du compteur électrique sur le réseau de distribution d'électricité
MPDU	MAC Protocol Data Unit (Unité de données de protocole MAC)
MSB	Most Significant Bit (Bit de poids fort)
MSDU	MAC Service Data Unit (Unité de données de service MAC)
MT	Mobile Termination (Terminaison mobile)
NB	Narrow-Band (Bande étroite)
ND	Neighbour Discovery (Découverte de voisins)
NRSRP	Narrowband Reference Signal Received Power (Puissance reçue du signal de référence à bande étroite)
NRSRQ	Narrowband Reference Signal Received Qualité (Qualité reçue du signal de référence à bande étroite)
NTP	Network Time Protocol (Protocole de synchronisation réseau)
F	Optional (Facultatif)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abréviation	Signification
OBIS	OBject Identification System (Système d'identification des objets)
OFDM	Orthogonal Frequency Division Multiplexing (Multiplexage par répartition orthogonale de la fréquence)
OTA	Over the Air (Par liaison radio) – Fait référence à la mise à jour de progiciel à l'aide de ZigBee®
PAN	Personal Area Network (Réseau local personnel) (terme utilisé en lien avec G3-PLC [1] et ZigBee®)
Pad	Padding (Remplissage)
PAP	Password Authentication Protocol (Protocole d'authentification par mot de passe)
PCLK	Pre-Configured Link Key (Clé de liaison préconfigurée) (terme ZigBee®)
PDU	Protocol Data Unit (Unité de données de protocole)
PhL, PHY	Physical Layer (Couche physique)
PIB	PLC Information Base (Base d'informations PLC)
PID	Program Identifier (Identifiant de programme)
PIN	Personal Identity Number (Numéro d'identification personnel)
PLC	Power Line Carrier (Courant porteur en ligne)
PLMN	Public Land Mobile Network (Réseau mobile terrestre public)
PNPDU	Promotion Needed PDU (PDU nécessaire pour la promotion)
POS	Point Of Sale (Point de vente) (comptage à paiement)
POS	Personal Operating Space (Espace de fonctionnement personnel) (ZigBee ®)
PPDU	Physical Protocol Data Unit (Unité de données de protocole de couche physique)
PPP	Point-to-Point Protocol (Protocole point à point)
PSTN	Public Switched Telephone Network (Réseau téléphonique public commuté)
QoS	Quality of Service (Qualité de service)
RB	Radio Band (Bande radio)
REJ PDU	Reject Protocol Data Unit (Unité de données de protocole de rejet)
RFC	Request for Comments (Appel à commentaires) – Document publié par l'Internet Engineering Task Force
RFD	Reduced Function Device (Dispositif à fonction réduite)
ROHC	Robust Header Compression (Compression robuste des en-têtes)
RREP	Route Reply (Réponse de route)
RREQ	Route Request (Demande de route)
RRER	Route Error (Erreur de route)
RSRQ	Reference Signal Received Quality (Qualité reçue du signal de référence)
RSRP	Reference Signal Received Power (Puissance reçue du signal de référence)
RSSI	Received Signal Strength Indication (Indication de l'intensité du signal reçu) (terme ZigBee®)
SAP	Service Access Point (Point d'accès au service)
SAS	Startup Attribute Set (Ensemble d'attributs de démarrage) (terme ZigBee®)
SCP	Shared Contention Period (Période de contention partagée)
SDU	Service Data Unit (Unité de données de service)
SE	Smart Energy (Réseau électrique intelligent)
SEP	Smart Energy Profile (Profil de réseau électrique intelligent) (terme ZigBee®)
S-FSK	Spread – Frequency Shift Keying (Modulation par déplacement de fréquence étalée)
SGERG88	Méthode de calcul de la compressibilité (compteur de gaz)
SHA	Secure Hash Algorithm (Algorithme de hachage sécurisé)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Abréviation	Signification
SID	Switch identifier (Identifiant de commutation)
SMS	Short Message Service (Service de messages courts)
SMTP	Simple Mail Transfer Protocol (Protocole de transfert de courrier simple)
SN	Short Name (Nom court)
SNA	SubNetwork Address (Adresse de sous-réseau)
SSAS	Service Specific Adaptation Sublayer (Sous-couche d'adaptation spécifique au service)
SSCS	Service Specific Convergence Layer (Couche de convergence spécifique au service)
StoC	Server to Client Challenge (Défi du serveur au client)
TAB	Dans le cas des profils EURIDIS sans DLMS® et sans DLMS®/COSEM: code de données. Dans le cas des profils utilisant DLMS® ou DLMS®/COSEM: valeur à laquelle l'appareil est sensibilisé au Discover
TABi	List of TAB field (Liste de champ TAB)
TCC	Transmission Control Code (Code de contrôle de transmission) (IPv4)
TCP	Transmission Control Protocol (Protocole de contrôle de transmission)
TFTP	Trivial File Transfer Protocol (Protocole simplifié de transfert de fichiers)
TOU	Time Of Use (Temps d'utilisation)
TTL	Time To Live (Durée de vie)
UDP	User Datagram Protocol (Protocole de datagramme utilisateur)
UMTS	Universal Mobile Telecommunications System (Système universel de télécommunications mobiles)
UNC	Unconfigured (Non configuré) (profil S-FSK PLC)
TUC	Coordinated Universal Time (Temps Universel Coordonné)
VIB	Value Information Block (Bloc d'informations sur les valeurs) (M-Bus)
VIF	Value Information Field (Champ d'informations sur les valeurs) (M-Bus)
VZ	Compteur de périodes de facturation (de l'allemand *VorwertZähler*, voir DIN 43863-3)
wake-up	déclenche le compteur à connecter au réseau de communication mis à la disposition du client (HES, par exemple)
WAN	Wide Area Network (Réseau étendu)
wM-Bus	Wireless M-Bus (M-Bus sans fil)
ZTC	ZigBee® Trust Center (Tiers de confiance ZigBee®)
[1)] Dans le cas de la technologie G3-PLC, le réseau PAN peut être défini comme un réseau local PLC.	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4 Classes d'interfaces COSEM

4.1 Principes de base

4.1.1 Généralités

Le présent paragraphe 4.1 décrit les principes de base sur lesquels sont construites les classes d'interfaces (IC) COSEM. Il donne aussi un aperçu de la manière d'utiliser les objets d'interface – instanciations des IC – à des fins de communication. Les systèmes de collecte de données et les équipements de comptage provenant de différents fournisseurs, conformes à ces spécifications, peuvent échanger des données d'une manière interopérable.

Pour les besoins de la spécification, la présente Norme utilise la technique de modélisation des objets.

Un objet est un ensemble d'attributs et de méthodes. Les attributs représentent les caractéristiques d'un objet. La valeur d'un attribut peut affecter le comportement d'un objet. Le premier attribut d'un objet est le *logical_name*(nom logique). Il fait partie de l'identification de l'objet. Un objet peut proposer un certain nombre de méthodes pour examiner ou modifier les valeurs des attributs.

Les objets qui partagent des caractéristiques communes sont généralisés en une classe d'interfaces, identifiée par un class_id (identifiant de classe). Au sein d'une IC spécifique, les caractéristiques communes (attributs et méthodes) sont décrites une seule fois pour tous les objets. Les instanciations des IC sont appelées "objets d'interface COSEM".

Les fabricants peuvent ajouter des méthodes et des attributs propriétaires à n'importe quel objet (voir 4.1.2).

La Figure 2 illustre ces termes par un exemple:

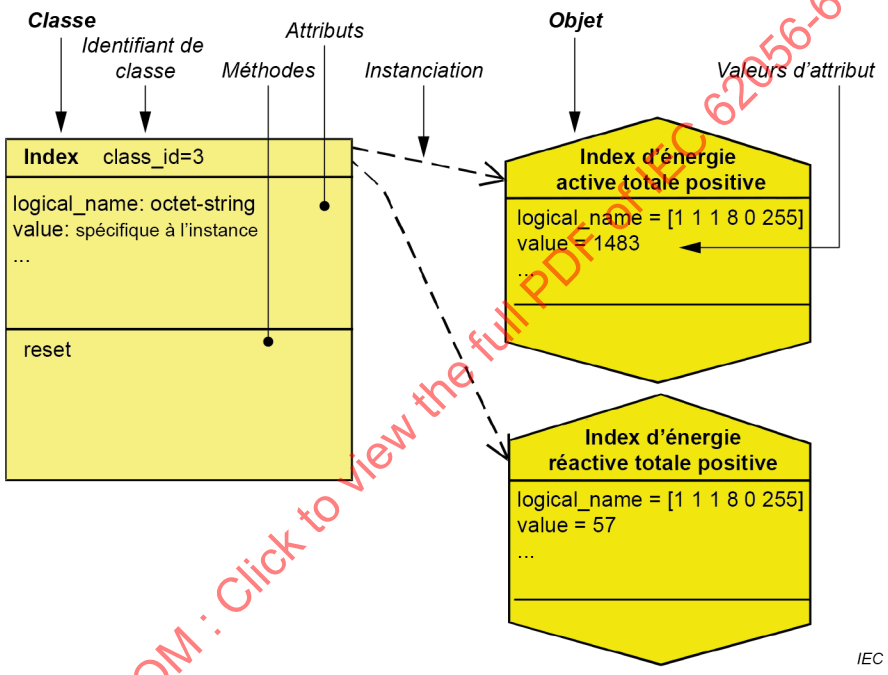


Figure 2 – Classe d'interfaces et ses instances

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

L'IC "Register" est formée en combinant les caractéristiques nécessaires à la modélisation du comportement d'un registre générique (contenant des informations mesurées ou statiques) du point de vue du client (système de collecte de données, terminal portable). Le contenu du registre est identifié par l'attribut *logical_name*. Le *logical_name* contient un identifiant OBIS (voir IEC 62056-6-1:2023). Le contenu (dynamique) réel du registre est porté par son attribut *value*.

La définition d'un compteur spécifique implique de définir plusieurs objets spécifiques. Dans l'exemple de la Figure 2, le compteur contient deux registres, c'est-à-dire que deux instances spécifiques de l'IC "Register" sont instanciées. Par le biais de l'instanciation, un objet COSEM devient un "registre d'énergie active, positive, totale" alors que l'autre devient un "registre d'énergie réactive, positive, totale".

NOTE Les objets d'interface COSEM (instances des IC COSEM) représentent le comportement du compteur tel que vu de "l'extérieur". Par conséquent, la modification de la valeur d'un attribut (la réinitialisation de l'attribut *value* d'un registre, par exemple) est toujours déclenchée de l'extérieur. Les modifications des attributs déclenchées de l'intérieur (la mise à jour de l'attribut *value* d'un registre, par exemple) ne sont pas décrites dans le présent modèle.

4.1.2 Méthodes de référencement

Les attributs et les méthodes des objets COSEM peuvent être référencés de deux manières différentes:

À l'aide de noms logiques (référencement LN): Dans ce cas, les attributs et les méthodes sont référencés via l'identifiant de l'instance d'objet COSEM à laquelle ils appartiennent.

La référence pour un attribut est: class_id, valeur de l'attribut *logical_name*, attribute_index.

La référence pour une méthode est: class_id, valeur de l'attribut *logical_name*, method_index, où:

- attribute_index est utilisé comme identifiant de l'attribut exigé. Les indices d'attribut sont spécifiés dans la définition de chaque IC. Ce sont des nombres positifs commençant à 1. Des attributs propriétaires peuvent être ajoutés: ils doivent être identifiés par des nombres négatifs;
- method_index est utilisé comme identifiant de la méthode exigée. Les indices de méthode sont spécifiés dans la définition de chaque IC. Ce sont des nombres positifs commençant à 1. Des méthodes propriétaires peuvent être ajoutées: elles doivent être identifiées par des nombres négatifs.

À l'aide de noms courts (référencement SN): Cette sorte de référencement est destinée à être utilisée dans des dispositifs simples. Dans ce cas, chaque attribut et chaque méthode d'un objet COSEM sont identifiés par un nombre entier de 13 bits. La syntaxe pour le nom court est la même que celle du nom d'une variable nommée selon la DLMS®. Voir IEC 61334-4-41:1996 et IEC 62056-5-3:2023, Article 8.

4.1.3 Base_name réservés pour des objets COSEM spéciaux

Afin de faciliter l'accès aux dispositifs à l'aide du référencement SN, certains short_name sont réservés sous forme de base_name pour des objets COSEM spéciaux. La plage pour les base_name réservés est comprise entre 0xFA00 et 0xFFF8. Les base_name spécifiques suivants sont définis (voir Tableau 1).

Tableau 1 – Base_name réservés pour le référencement SN

Base_name (objectName)	Objet COSEM
0xFA00	Association SN
0xFB00	Script table (instanciation: Broadcast "Script table")
0xFC00	SAP assignment

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

0xFD00	Objet "Data" ou "Register" contenant le "COSEM logical device name" (nom d'appareil logique COSEM) dans l'attribut "value"

4.1.4 Notation pour la description des classes

4.1.4.1 Généralités

Le présent paragraphe décrit la notation utilisée pour définir les IC.

Un texte court décrit la fonctionnalité et l'application de l'IC. Le Tableau 2 donne une vue d'ensemble de l'IC, y compris le nom de classe, les attributs et les méthodes. Chaque attribut et chaque méthode doivent être décrits dans le détail. Le modèle est présenté ci-dessous.

Tableau 2 – Vue d'ensemble de la classe d'interfaces

Nom de la classe		**Cardinalité**	**class_id, version**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. …	(…)	…				x + 0x…
3. …	(…)	…				x + 0x…
Méthodes spécifiques (si exigées)		***o/f***				
1.		…				x + 0x…
2.		…				x + 0x…
3.		…				x + 0x…

4.1.4.2 Nom de la classe

Décrit la classe d'interfaces ("Register", "Clock", "Profile generic"…, par exemple).

NOTE Les noms des classes d'interfaces sont indiqués entre guillemets.

4.1.4.3 Cardinalité

Spécifie le nombre d'instances de l'IC au sein d'un dispositif logique (voir 4.1.8).

value L'IC doit être instanciée exactement "value" fois.

min…max L'IC doit être instanciée au moins "min." fois et au plus "max." fois. Si min. est nul (0), alors l'IC est facultative; sinon (min. > 0) "min." instanciations de l'IC sont obligatoires.

4.1.4.4 class_id

Code d'identification de l'IC (plage de 0 à 65 535). Le class_id de chaque objet est récupéré avec le nom logique par lecture de l'attribut object_list d'un objet "Association LN"/"Association SN".

– Les class_id compris entre 0 et 8 191 sont réservés pour être spécifiés par la DLMS® UA.
– Les class_id compris entre 8 192 et 32 767 sont réservés pour des IC spécifiques à un fabricant.
– Les class_id compris entre 32 768 et 65 535 sont réservés pour des IC spécifiques à un groupe d'utilisateurs.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

La DLMS® UA réserve le droit d'attribuer les plages à des fabricants individuels ou à des groupes d'utilisateurs.

Les class_ID compris entre 32 768 et 32 778 sont alloués à l'Association STS.

4.1.4.5 version

Code d'identification de la version de l'IC. La version de chaque objet est récupérée avec le class_id et le nom logique par lecture de l'attribut object_list d'un objet "Association LN"/"Association SN".

Au sein d'un même dispositif logique, toutes les instances d'une certaine IC doivent être de la même version.

4.1.4.6 Attributs

Spécifie les attributs qui appartiennent à l'IC.

(dyn.) Classifie un attribut qui porte une valeur de processus, qui est mis à jour par le compteur lui-même.

(static) Classifie un attribut qui n'est pas mis à jour par le compteur lui-même (données de configuration, par exemple).

NOTE 1 Certains attributs peuvent être soit statiques, soit dynamiques en fonction de l'application. Dans ces cas, cette propriété n'est pas indiquée.

NOTE 2 Les noms d'attribut utilisent un trait de soulignement. Lorsqu'ils sont mentionnés dans le texte, ils apparaissent en *italique*. Exemple: *logical_name*.

4.1.4.7 logical_name

octet-string Il s'agit toujours du premier attribut d'une IC. Il identifie l'instanciation (objet COSEM) de cette IC. La valeur du *logical_name* est conforme à l'OBIS (voir Article 6 et IEC 62056-6-1:2023).

4.1.4.8 Type de données

Définit le type de données d'un attribut (voir 4.1.5).

4.1.4.9 Min.

Spécifie si l'attribut a une valeur minimale.

x L'attribut a une valeur minimale.

<*vide*> L'attribut n'a pas de valeur minimale.

4.1.4.10 Max.

Définit si l'attribut a une valeur maximale.

x L'attribut a une valeur maximale.

<*vide*> L'attribut n'a pas de valeur maximale.

4.1.4.11 Def.

Spécifie si l'attribut a une valeur par défaut. Il s'agit de la valeur de l'attribut après réinitialisation.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

x L'attribut a une valeur par défaut.

<vide> La valeur par défaut n'est pas définie par la spécification de l'IC.

4.1.4.12 Nom court

Lorsque le référencement par nom court (SN) est utilisé, chaque attribut et chaque méthode des instances d'un objet doivent être mis en correspondance avec des noms courts.

Le base_name x de chaque instance d'objet est la variable nommée selon la DLMS® avec laquelle l'attribut logical_name est mis en correspondance. Il est sélectionné pendant la phase d'implémentation. La définition de l'IC spécifie les décalages pour les autres attributs et pour les méthodes.

4.1.4.13 Méthodes spécifiques

Fournit une liste de méthodes spécifiques qui appartiennent à l'objet.

Nom de méthode () La méthode doit être décrite dans la sous-section "Description de méthode".

NOTE Les noms de méthodes utilisent un trait de soulignement. Lorsqu'ils sont mentionnés dans le texte, ils apparaissent en *italique*. Exemple: *add_object*.

4.1.4.14 o/f

Définit si la méthode est obligatoire ou facultative.

o (obligatoire) La méthode est obligatoire.

f (facultatif) La méthode est facultative.

4.1.4.15 Description d'attribut

Décrit chaque attribut avec son type de données (si le type de données n'est pas simple), son format de données et ses propriétés (valeurs minimale, maximale et par défaut).

4.1.4.16 Description de la méthode

Décrit chaque méthode et le comportement invoqué des objets COSEM instanciés.

NOTE Les services permettant d'accéder aux attributs ou aux méthodes par le protocole sont spécifiés dans l'IEC 62056-5-3:2023, de 6.6 à 6.17.

4.1.4.17 Accès sélectif

Les services liés à l'attribut xDLMS font généralement référence à l'ensemble de l'attribut. Cependant, pour certains attributs, un accès sélectif à seulement une partie de l'attribut peut être assuré. La partie de l'attribut est identifiée par des paramètres d'accès sélectif spécifiques, qui sont définis comme partie intégrante de la spécification de l'attribut.

L'accès sélectif est disponible avec les méthodes et les attributs de classe d'interfaces suivants:

- objets "Profile generic", attribut **buffer**;
- objets "Association SN", attributs **object_list** et **access_rights_list**;
- objets "Association LN", attribut **object_list**;
- objets "Compact data", attribut **compact_buffer**;
- objets "Push", attribut **push_object_list**;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- objets "Data protection", attribut **protection_object_list**, méthodes **get_protected_attributes** et **set_protected_attributes**.

4.1.5 Types de données communs

Le Tableau 3 contient la liste des types de données utilisables pour les attributs des objets COSEM.

Tableau 3 – Types de données communs

Description de type	Éti-quette[a]	Définition	Plage de valeurs
-- types de données simples			
null-data	[0]		
boolean	[3]	boolean	TRUE ou FALSE
bit-string	[4]	Séquence ordonnée de valeurs booléennes	
double-long	[5]	Integer32	-2 147 483 648... 2 147 483 647
double-long-unsigned	[6]	Unsigned32	0...4 294 967 295
	[7]	Étiquette du type "floating-point" (virgule flottante) dans l'IEC 61334-4-41:1996, non utilisable dans DLMS®/COSEM. Voir étiquettes [23] et [24]	
octet-string	[9]	Séquence ordonnée d'octets (octets de 8 bits)	
visible-string	[10]	Séquence ordonnée de caractères ASCII	
	[11]	Étiquette du type "time" (temps) dans l'IEC 61334-4-41:1996, non utilisable dans DLMS®/COSEM. Voir étiquette [27]	
utf8-string	[12]	Séquence ordonnée de caractères codés en UTF-8	
bcd	[13]	binary coded decimal (décimal codé binaire)	
integer	[15]	Integer8	-128...127
long	[16]	Integer16	-32 768...32 767
unsigned	[17]	Unsigned8	0...255
long-unsigned	[18]	Unsigned16	0...65 535
long64	[20]	Integer64	$-2^{63}...2^{63}-1$
long64-unsigned	[21]	Unsigned64	$0...2^{64}-1$
enum	[22]	Les éléments du type "enumeration" (énumération) sont définis dans la section "*Description d'attribut*" ou "*Description de la méthode*" d'une spécification d'IC COSEM.	0...255
float32	[23]	OCTET STRING (SIZE(4))	Pour la mise en forme, voir 4.1.6.2.
float64	[24]	OCTET STRING (SIZE(8))	
date-time	[25]	OCTET STRING (SIZE(12))	Pour la mise en forme, voir 4.1.6.1.
date	[26]	OCTET STRING (SIZE(5))	
time	[27]	OCTET STRING (SIZE(4))	
delta-integer	[28]	Integer8	-128...127
delta-long	[29]	Integer16	-32 768...32 767
delta-double-long	[30]	Integer32	-2 147 483 648... 2 147 483 647
delta-unsigned	[31]	Unsigned8	0...255
delta-long-unsigned	[32]	Unsigned16	0...65 535

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Description de type	Étiquette[a]	Définition	Plage de valeurs
delta-double-long-unsigned	[33]	Unsigned32	0…4 294 967 295
-- types de données complexes			
array	[1]	Les éléments du type "array" (tableau) sont définis dans la section "*Attribut*" ou "*Description de la méthode*" d'une spécification d'IC COSEM.	
structure	[2]	Les éléments du type "structure" sont définis dans la section "*Description de la méthode*" d'une spécification d'IC COSEM.	
compact-array	[19]	Offre une alternative de codage compact des données complexes.	
-- CHOICE		Pour certains attributs des objets d'interface COSEM, le type de données peut être choisi au moment de l'instanciation, pendant la phase d'implémentation du serveur COSEM. Le serveur doit toujours renvoyer le type de données et la valeur de chaque attribut, ce qui, avec le nom logique, assure une interprétation non ambiguë. La liste des types de données possibles est définie dans la section "*Description d'attribut*" dans une spécification d'IC COSEM.	
[a] Les étiquettes sont définies dans l'IEC 62056-5-3:2023, Article 8.			

4.1.6 Formats de données

4.1.6.1 Formats des date et heure

Les informations de date et d'heure peuvent être représentées à l'aide du type de données *octet-string*.

NOTE 1 Dans ce cas, le codage inclut l'étiquette du type de données *octet-string*, la longueur de la chaîne d'octets et les éléments de *date*, *heure* et/ou *date-heure*, selon le cas.

Les informations de date et d'heure peuvent également être représentées à l'aide des types de données *date*, *time* et *date-time*.

NOTE 2 Dans ces cas, le codage inclut uniquement l'étiquette des types de données *date*, *time* ou *date-time*, selon le cas, et les éléments de date, d'heure ou date-heure.

NOTE 3 Les spécifications (SIZE ( )) sont applicables uniquement lorsque la date, l'heure ou la date-heure sont représentées par les types de données *date*, *time* ou *date-time*.

date OCTET STRING (SIZE(5))

{
year highbyte,
year lowbyte,
month,
day of month,
day of week
}

Où:

year (année): interprété comme un long-unsigned
plage 0…big
0xFFFF = non spécifié

year highbyte et year lowbyte représentent les 2 octets du long-unsigned

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

month (mois): interprété comme un unsigned
plage 1…12, 0xFD, 0xFE, 0xFF
1 correspond à janvier
0xFD = daylight_savings_end
0xFE = daylight_savings_begin
0xFF = non spécifié

dayOfMonth (jour du mois): interprété comme un unsigned
plage 1…31, 0xFD, 0xFE, 0xFF
0xFD = avant-dernier jour du mois
0xFE = dernier jour du mois
0xE0 à 0xFC = réservés
0xFF = non spécifié

dayOfWeek (jour de la semaine): interprété comme un unsigned
plage 1…7, 0xFF
1 correspond à lundi
0xFF = non spécifié

Pour les dates répétitives, les parties non utilisées doivent être mises à "non spécifié".

Pour les pays qui n'utilisent pas le calendrier grégorien, "Month 1" est le premier mois du calendrier et la plage de dayOfMonth peut être différente.

Les éléments dayOfMonth et dayOfWeek doivent être interprétés ensemble:

– si le dernier dayOfMonth est spécifié (0xFE) et que dayOfWeek est un caractère générique, cela spécifie le dernier jour calendaire du mois;
– si le dernier dayOfMonth est spécifié (0xFE) et qu'un dayOfWeek explicite est spécifié (par exemple 7, dimanche), il s'agit de la dernière occurrence du weekday (jour de la semaine) spécifié dans le month (mois), c'est-à-dire le dernier dimanche;
– si year (année) n'est pas spécifié (0xFFFF) et que dayOfMonth et dayOfWeek sont tous deux explicitement spécifiés, il doit être interprété comme le dayOfWeek du dayOfMonth ou suivant celui-ci;
– si year (année) et month (mois) sont spécifiés et que dayOfMonth et dayOfWeek sont tous deux explicitement spécifiés, mais que les valeurs ne sont pas compatibles, cela doit être considéré comme étant une erreur.

Exemples:

1) year = 0xFFFF, month = 0x FF, dayOfMonth = 0xFE, dayofWeek = 0xFF: dernier jour du mois de chaque année et mois;

2) year = 0xFFFF, month = 0x FF, dayOfMonth = 0xFE, dayofWeek = 0x07: dernier dimanche de chaque année et mois;

3) year = 0xFFFF, month = 0x03, dayOfMonth = 0xFE, dayofWeek = 0x07: dernier dimanche de mars de chaque année;

4) year = 0xFFFF, month = 0x03, dayOfMonth = 0x01, dayofWeek = 0x07: premier dimanche de mars de chaque année;

5) year = 0xFFFF, month = 0x03, dayOfMonth = 0x16, dayofWeek = 0x05: quatrième vendredi de mars de chaque année;

6) year = 0xFFFF, month = 0x0A, dayOfMonth = 0x16, dayofWeek = 0x07: quatrième dimanche d'octobre de chaque année;

7) year = 0x07DE, month = 0x08, dayOfMonth = 0x13, (2014.08.13, Wednesday) dayofWeek = 0x02 (Tuesday): erreur, étant donné que le dayOfMonth et le dayOfWeek de l'année et du mois donnés ne correspondent pas.

time OCTET STRING (SIZE(4))

{

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

hour,
minute,
second,
hundredths
}

Où:

hour (heure): interprété comme un unsigned
plage 0…23, 0xFF;

minute: interprété comme un unsigned
plage 0…59, 0xFF;

second (seconde): interprété comme un unsigned
plage 0…59, 0xFF;

hundredths (centièmes): interprété comme un unsigned
plage 0…99, 0xFF.

Pour hour, minute, second et hundredths: 0xFF = non spécifié.

Pour les heures répétitives, les parties non utilisées doivent être mises à "non spécifié".

date-time OCTET STRING (SIZE(12))
{
year highbyte,
year lowbyte,
month,
day of month,
day of week,
hour,
minute,
second,
hundredths of second,
deviation highbyte,
deviation lowbyte,
clock status
}

Les éléments de *date* et *heure* sont codés comme indiqué ci-dessus. Certains peuvent être mis à "non spécifié" comme indiqué ci-dessus.

De plus:

deviation: interprété comme un long
plage -720…+720 en minutes de l'heure locale par rapport à l'heure au TUC
0x8000 = non spécifié

deviation highbyte et deviation lowbyte représentent les 2 octets du "long".

clock_status est interprété comme un unsigned.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Les bits sont définis comme suit:

bit 0 (LSB): valeur non valide [a];
bit 1: valeur douteuse [b];
bit 2: base d'horloge différente [c];
bit 3: statut d'horloge non valide [d];
bit 4: réservé;
bit 5: réservé;
bit 6: réservé;
bit 7 (MSB): heure d'été/hiver active [e].

0xFF = non spécifié

[a] L'heure ne pouvait pas être récupérée après un incident. Les conditions détaillées sont spécifiques au fabricant (après une coupure de l'alimentation de l'horloge, par exemple). Pour un statut valide, le bit 0 ne doit pas être défini si le bit 1 l'est.

[b] L'heure pouvait être récupérée après un incident, mais la valeur ne peut pas être garantie. Les conditions détaillées sont spécifiques au fabricant. Pour un statut valide, le bit 1 ne doit pas être défini si le bit 0 l'est.

[c] Le bit est positionné si les informations de temps élémentaires pour l'horloge à l'instant effectif proviennent d'une source de temps différente de la source spécifiée dans clock_base.

[d] Ce bit indique qu'au moins un bit du statut de l'horloge est non valide. Certains bits peuvent être corrects. La signification exacte doit être expliquée dans la documentation du fabricant.

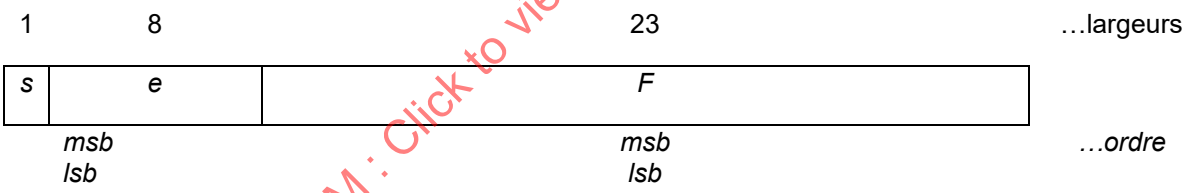
[e] Fanion mis à "true" (vrai): l'heure transmise contient l'écart été/hiver (heure d'été).

Fanion mis à "false" (faux): l'heure transmise ne contient pas l'écart été/hiver (heure normale).

4.1.6.2 Formats de nombres en virgule flottante

Les formats de nombres en virgule flottante sont définis dans l'ISO/IEC 60559:2020.

Le format "single" (simple) est:



où:

s est le bit de signe;

e est l'exposant; il a une largeur de 8 bits et le biais d'exposant est +127;

f est la fraction, sa taille est de 23 bits.

Avec ces valeurs, la valeur est (si 0 < e < 255):

$$\upsilon = (-1)^s \times 2^{e-127} \times (1 \times f)$$

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Le format "double" est:

1	11	52	…largeurs
s	*e*	*F*	
	msb *lsb*	*msb* *lsb*	*…ordre*

où:

s est le bit de signe;

e est l'exposant; il a une largeur de 11 bits et le biais d'exposant est +1 023;

f est la fraction, sa taille est de 52 bits.

Avec ces valeurs, la valeur est (si 0 < e < 2 047):

$$\upsilon = (-1)^s \times 2^{e-1023} \times (1 \times f)$$

Pour plus de détails, voir ISO/IEC 60559:2020.

Les nombres en virgule flottante doivent être représentés sous la forme d'une chaîne d'octets de longueur fixe, contenant les 4 octets (float32) du nombre en virgule flottante de format "single" ou les 8 octets (float64) du nombre en virgule flottante de format "double" tels que spécifiés ci-dessus, l'octet de poids fort en premier.

EXEMPLE 1 La valeur décimale "1" représentée au format à virgule flottante "single" est:

Bit 31 Bit de signe **0** 0: + 1: -	Bits 30 à 23 Champ exposant: **01111111** Valeur décimale du champ exposant et exposant: **127**-127 = 0	Bits 22-0 Significande 1.**00000000000000000000000** Valeur décimale du significande: 1,0000000

NOTE Le significande est le nombre binaire 1 suivi de la séparation fractionnaire suivie des bits binaires de la fraction.

Le codage, y compris l'étiquette du type de données est (toutes les valeurs sont hexadécimales): 17 3F 80 00 00.

EXEMPLE 2 La valeur décimale "1" représentée au format à virgule flottante "double" est:

Bit 63 Bit de signe **0** 0: + 1: -	Bits 62 à 52 Champ exposant: **01111111111** Valeur décimale du champ exposant et exposant: **1023**-1023 = 0	Bits 51-0 Significande 1.**0000000000000000000000000000000000000000000000000000** Valeur décimale du significande: 1.0000000000000000

Le codage, y compris l'étiquette du type de données est (toutes les valeurs sont hexadécimales): 18 3F F0 00 00 00 00 00 00.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

EXEMPLE 3 La valeur décimale "62056" représentée au format à virgule flottante "single" est:

Bit 31 Bit de signe **0** 0: + 1: -	Bits 30 à 23 Champ exposant: **10001110** Valeur décimale du champ exposant et exposant: **142**-127 = 15	Bits 22-0 Significande 1.**11100100110100000000000** Valeur décimale du significande: 1,8937988

Le codage, y compris l'étiquette du type de données est (toutes les valeurs sont hexadécimales): 17 47 72 68 00.

EXEMPLE 4 La valeur décimale "62056" représentée au format à virgule flottante "double" est:

Bit 63 Bit de signe **0** 0: + 1: -	Bits 62 à 52 Champ exposant: **10000001110** Valeur décimale du champ exposant et exposant: **1038**-1023 = 15	Bits 51-0 Significande 1.**11100100110100000000000000000000000000000000000000000000** Valeur décimale du significande: 1,8937988281250000

Le codage, y compris l'étiquette du type de données est (toutes les valeurs sont hexadécimales): 18 40 EE 4D 00 00 00 00 00.

4.1.6.3 Codage de données nulles et de valeur delta

Le codage de données nulles et le codage de valeur delta sont deux méthodes qui permettent d'accroître l'efficacité lorsqu'un ou plusieurs tableaux de structures doivent être transférés, par exemple l'attribut buffer de l'IC "Profile generic" ou l'attribut table_cell_values de l'IC "Register table".

Avec le codage de données nulles, la valeur peut être remplacée par "null-data" si elle peut être récupérée sans ambiguïté à partir de la valeur précédente (par exemple pour le temps: si elle peut être calculée à partir de la valeur précédente et de capture_period ou pour une valeur: si elle est égale à la valeur précédente).

Avec le codage de valeur delta, la valeur d'origine peut être remplacée par une valeur delta comme suit:

- la première valeur doit être codée en utilisant un type de données integer ou unsigned approprié;
- les valeurs suivantes doivent être codées de la même manière, ou elles peuvent être remplacées par une valeur delta codée en utilisant un type de données delta approprié.

Les deux méthodes peuvent être utilisées individuellement ou combinées et elles peuvent être utilisées avec compression pour améliorer encore l'efficacité.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.1.7 Modèle de serveur COSEM

Le serveur COSEM est structuré en trois niveaux hiérarchiques (voir Figure 3):

Niveau 1: Dispositif physique
Niveau 2: Dispositif logique
Niveau 3: Objets COSEM accessibles

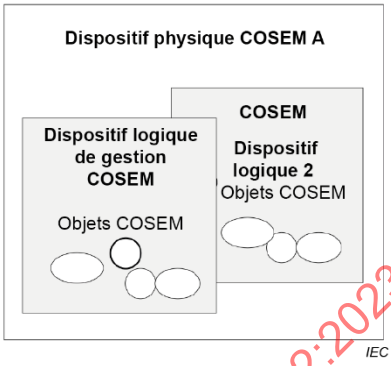


Figure 3 – Modèle de serveur COSEM

L'exemple de la Figure 4 montre la façon dont un dispositif de comptage combiné peut être structuré à l'aide du modèle de serveur COSEM.

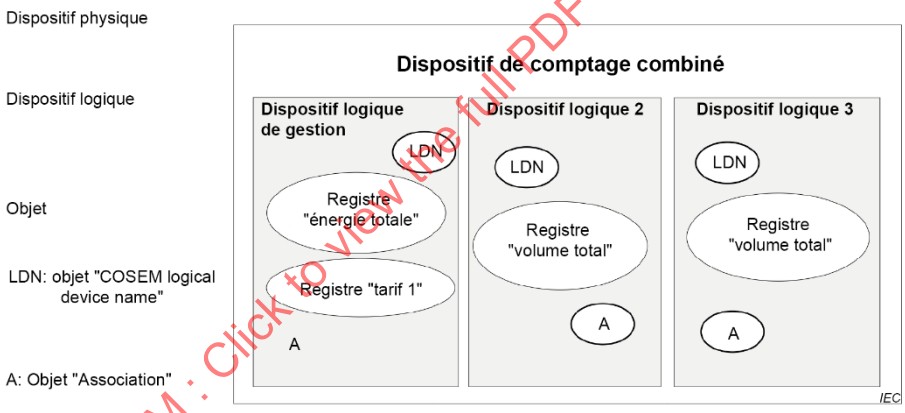


Figure 4 – Dispositif de comptage combiné

4.1.8 Dispositif logique COSEM

4.1.8.1 Généralités

Le dispositif logique COSEM contient un ensemble d'objets COSEM. Chaque dispositif physique doit contenir un "Dispositif logique de gestion".

L'adressage des dispositifs logiques COSEM doit être assuré par le plan d'adressage des couches inférieures de la pile protocolaire utilisée.

4.1.8.2 Nom de dispositif logique COSEM (LDN)

Le dispositif logique COSEM peut être identifié par son LDN COSEM unique. Ce nom peut être récupéré d'une instance de l'IC "SAP assignment" (voir 4.4.5) ou d'un objet COSEM "COSEM logical device name" (voir 6.2.35).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Le LDN est défini sous la forme d'une chaîne de 16 octets au maximum. Les trois premiers octets doivent porter l'identifiant du fabricant [2]. Le fabricant doit s'assurer que le LDN, commençant par les trois octets identifiant le fabricant et suivi de 13 octets au maximum, est unique.

4.1.8.3 "Vue des associations" du dispositif logique

Afin d'accéder aux objets COSEM dans le serveur, une association d'applications (AA) doit d'abord être établie avec un client. Les AA identifient les partenaires et caractérisent le contexte dans lequel les applications associées communiqueront. Les principales parties de ce contexte sont:

- le contexte d'application;
- le mécanisme d'authentification;
- le contexte xDLMS.

Les AA sont modélisées par des objets COSEM particuliers:

- les instances de l'IC "Association SN" (voir 4.4.3) sont utilisées avec le référencement par nom court;
- les instances de l'IC "Association LN" (voir 4.4.4) sont utilisées avec le référencement par nom logique.

En fonction de l'AA établie entre le client et le serveur, différents droits d'accès peuvent être accordés par le serveur. Les droits d'accès concernent un ensemble d'objets COSEM – les objets visibles – qui peuvent être accessibles ("vus") au sein d'une AA donnée. En outre, l'accès aux attributs et méthodes de ces objets COSEM peut aussi être limité au sein de l'AA (par exemple, un certain type de client ne peut lire qu'un attribut particulier d'un objet COSEM, mais ne peut pas l'écrire). Les droits d'accès peuvent également stipuler la protection cryptographique exigée.

Le client peut obtenir la liste des objets COSEM visibles – la "vue des associations" – en lisant l'attribut *object_list* de l'objet "Association" approprié.

4.1.8.4 Contenu obligatoire d'un dispositif logique COSEM

Les objets suivants doivent être présents dans chaque dispositif logique COSEM. Ils doivent être accessibles pour GET/Read dans toutes les AA avec ce dispositif logique:

- objet "COSEM logical device name";
- objet "Association" (LN ou SN) actuel.

Si l'objet "SAP Assignment" est présent, l'objet "COSEM logical device name" ne doit pas nécessairement être présent.

Pour identifier le progiciel, les objets suivants sont obligatoires:

- un objet d'identification du progiciel actif qui contient l'identifiant du progiciel actuellement actif;
- un objet de signature du progiciel actif qui contient la signature numérique du progiciel actuellement actif.

NOTE L'algorithme de signature numérique n'est pas spécifié ici.

[2] Administré par la DLMS® User Association, en coopération avec la FLAG Association.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si un dispositif logique dispose de plusieurs progiciels, un objet d'identification du progiciel actif et un objet de signature du progiciel actif doivent être présents pour chaque progiciel.

Les objets suivants peuvent éventuellement être présents:

- un ou plusieurs objets de version du progiciel actif qui contiennent la version du progiciel actuellement actif.

4.1.8.5 Dispositif logique de gestion

Comme spécifié en 4.1.8.1, le dispositif logique de gestion est un élément obligatoire de tout dispositif physique. Il a une adresse réservée. Il doit prendre en charge une AA avec un client public avec l'authentification au niveau de sécurité le plus bas (pas de sécurité). Son rôle est d'assurer la divulgation de la structure interne du dispositif physique et de prendre en charge la notification d'événements dans le serveur.

En plus de l'objet "Association" modélisant l'AA avec le client public, le dispositif logique de gestion doit contenir un objet "SAP assignment", donnant son SAP et le SAP de tous les autres dispositifs logiques au sein du dispositif physique. L'objet "SAP assignment" doit être lisible au moins par le client public.

S'il n'y a qu'un seul dispositif logique au sein du dispositif physique, l'objet "SAP assignment" peut être omis.

4.1.9 Sécurité des informations

DLMS®/COSEM fournit plusieurs fonctions de sécurité de l'information pour l'accès et le transport des données:

- la sécurité d'accès des données contrôle l'accès aux données détenues par un serveur DLMS®/COSEM;
- la sécurité de transport des données permet à la partie émettrice d'appliquer une protection cryptographique aux APDU de xDLMS envoyées. Cela exige des APDU chiffrées. La partie réceptrice peut retirer ou vérifier cette protection;
- la sécurité des données COSEM permet de protéger les valeurs d'attribut COSEM, ainsi que les paramètres d'appel et de retour de la méthode.

Pour une description de ces mécanismes de sécurité, voir IEC 62056-5-3:2023, Article 5.

La sécurité des informations est assurée au niveau de la couche application DLMS®/COSEM et au niveau de l'objet COSEM. Elle est prise en charge/gérée par les objets suivants:

- "Association SN" (voir 4.4.3);
- "Association LN" (voir 4.4.4);
- "Security setup" (voir 4.4.7);
- "Data protection" (voir 4.4.9).

4.2 Vue d'ensemble des classes d'interfaces COSEM

Les IC définies à l'heure actuelle et les relations entre ces classes sont représentées à la Figure 5, la Figure 6 et la Figure 7.

NOTE 1 La "base" d'IC proprement dite n'est pas spécifiée de façon explicite. Elle ne contient qu'un seul attribut *logical_name*.

NOTE 2 Dans la description des IC "Demand register", "Clock" et "Profile generic", les 2^{e} attributs sont étiquetés différemment de celui du 2^{e} attribut de l'IC "Data", à savoir *current_average_value*, *time* et *buffer* par opposition à *value*. Le but est de mettre l'accent sur la nature spécifique de *value*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE 3 Dans ces figures, les classes d'interfaces sont présentées dans chaque groupe en augmentant class_id. Dans les paragraphes spécifiant les différents groupes de classes d'interfaces, les nouvelles classes d'interfaces sont placées à la fin du paragraphe correspondant.

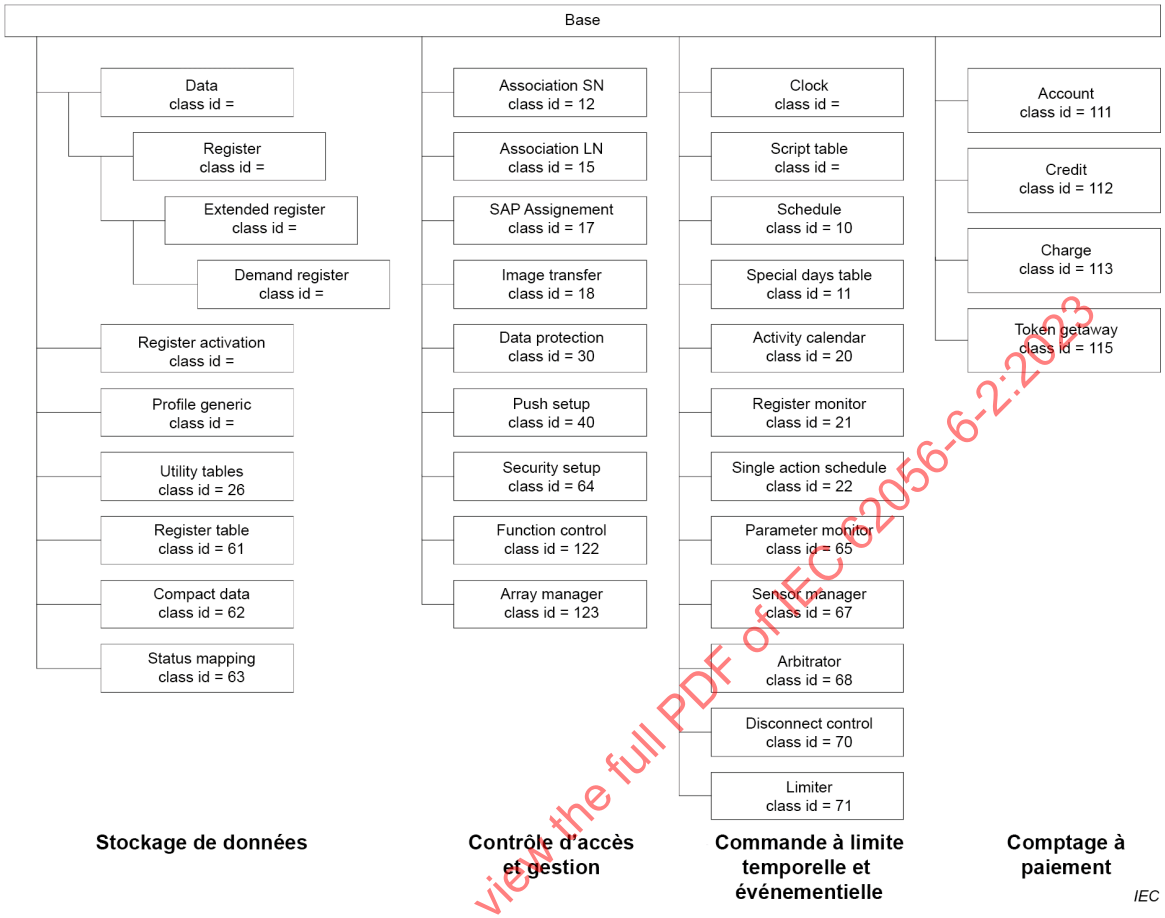
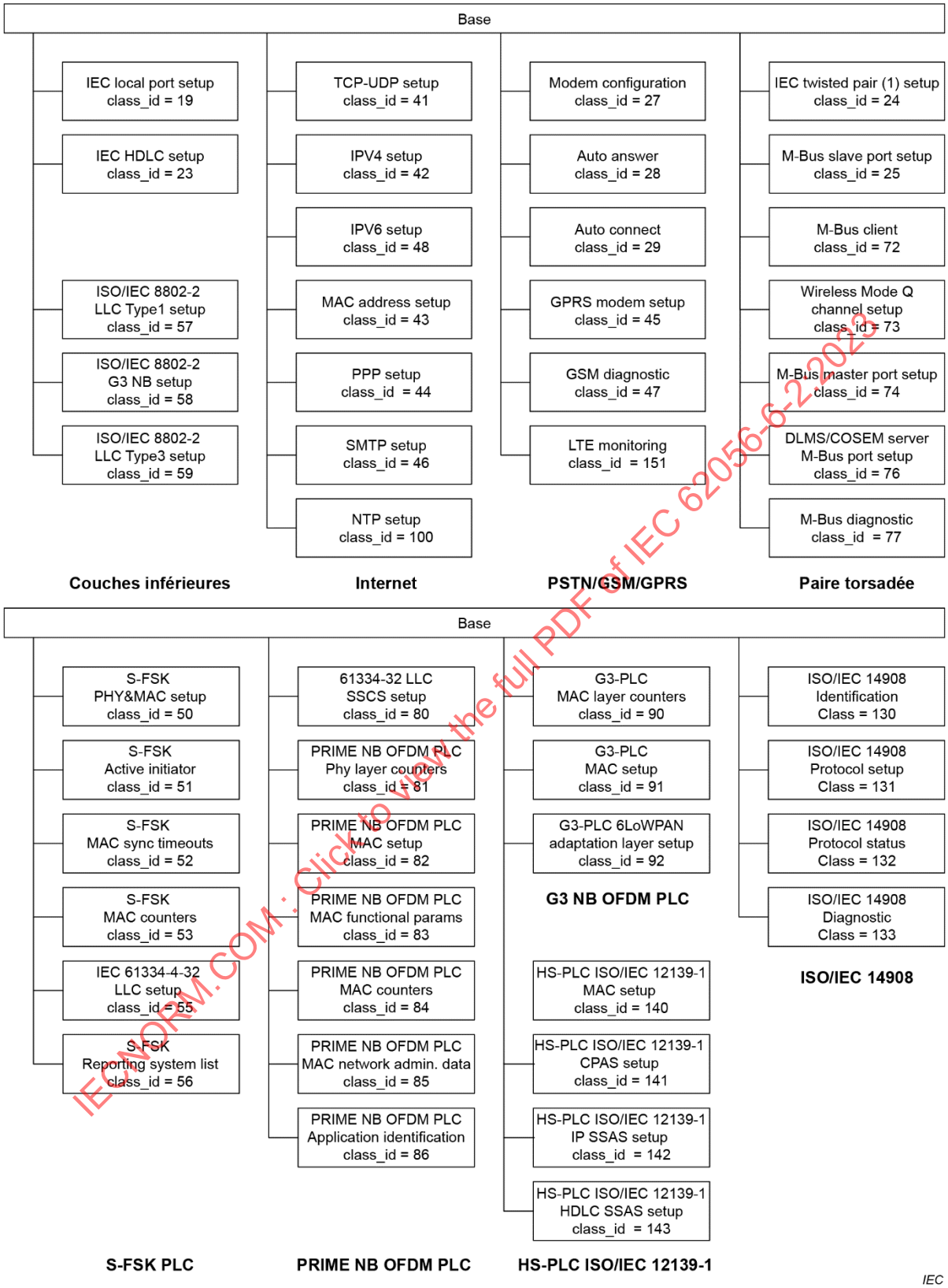


Figure 5 – Vue d'ensemble des classes d'interfaces – Partie 1

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 6 – Vue d'ensemble des classes d'interfaces – Partie 2

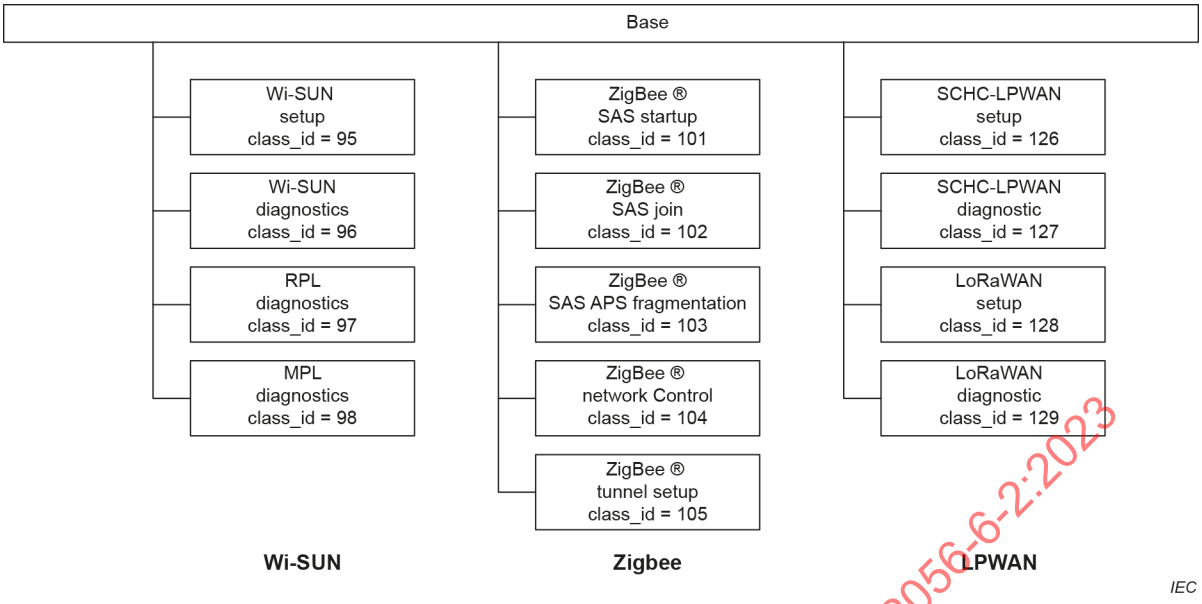


Figure 7 – Vue d'ensemble des classes d'interfaces – Partie 3

Le Tableau 4 répertorie les classes d'interfaces par class_id.

Tableau 4 – Liste des classes d'interfaces par class_id

Nom de la classe d'interface	class_id	version(s)	Paragraphe
Data	1	0	4.3.1
Register	3	0	4.3.2
Extended register	4	0	4.3.3
Demand register	5	0	4.3.4
Register activation	6	0	4.3.5
Profile generic	7	1 0	4.3.6 5.3.1
Clock	8	0	4.5.1
Script table	9	0	4.5.2
Schedule	10	0	4.5.3
Special days table	11	0	4.5.4
Association SN	12	4 3 2 1 0	4.4.3 5.4.4 5.4.3 5.4.2 5.4.1
Association LN	15	3 2 1 0	4.4.4 5.4.7 5.4.6 5.4.5
SAP Assignment	17	0	4.4.5
Image transfer	18	0	4.4.6
IEC local port setup	19	1 0	4.7.1 5.7.1
Activity calendar	20	0	4.5.5

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Nom de la classe d'interface	class_id	version(s)	Paragraphe
Register monitor	21	0	4.5.6
Single action schedule	22	0	4.5.7
IEC HDLC setup	23	1 0	4.7.2 5.7.2
IEC twisted pair (1) setup	24	1 0	4.7.3 5.7.3
M-BUS slave port setup	25	0	4.8.2
Utility tables	26	0	4.3.7
Modem configuration PSTN modem configuration	27	1 0	4.7.4 5.7.4
Auto answer	28	2 1 0	4.7.5 – 5.7.5
Auto connect PSTN Auto dial	29	2 1 0	4.7.6 5.7.7 5.7.6
Data protection	30	0	4.4.9
Push setup	40	2 1 0	4.4.8.2 5.4.10 5.4.9
TCP-UDP setup	41	0	4.9.1
IPv4 setup	42	0	4.9.2
MAC address setup (Ethernet setup)	43	0	4.9.4 4.12.10
PPP setup	44	0	4.9.5
GPRS modem setup	45	0	4.7.7
SMTP setup	46	0	4.9.6
GSM diagnostic	47	2 1 0	4.7.8 5.7.9 5.7.8
IPv6 setup	48	0	4.9.3
S-FSK Phy&MAC setup	50	1 0	4.10.3 5.10.1
S-FSK Active initiator	51	0	4.10.4
S-FSK MAC synchronization timeouts	52	0	4.10.5
S-FSK MAC counters	53	0	4.10.6
IEC 61334-4-32 LLC setup S-FSK IEC 61334-4-32 LLC setup	55	1 0	4.10.7 5.10.2
S-FSK Reporting system list	56	0	4.10.8
ISO/IEC 8802-2 LLC Type 1 setup	57	0	4.11.2
ISO/IEC 8802-2 LLC Type 2 setup	58	0	4.11.3
ISO/IEC 8802-2 LLC Type 3 setup	59	0	4.11.4
Register table	61	0	4.3.8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Nom de la classe d'interface	class_id	version(s)	Paragraphe
Compact data	62	1 0	4.3.10 5.3.2
Status mapping	63	0	4.3.9
Security setup	64	1 0	4.4.7 5.4.8
Parameter monitor	65	1 0	4.5.10 5.5.1
Sensor manager	67	0	4.5.11
Arbitrator	68	0	4.5.12
Disconnect control	70	0	4.5.8
Limiter	71	0	4.5.9
M-Bus client	72	1 0	4.8.3 5.8.1
Wireless Mode Q channel	73	0	4.8.4
M-Bus master port setup	74	0	4.8.5
DLMS®/COSEM server M-Bus port setup	76	0	4.8.6
M-Bus diagnostic	77	0	4.8.7
61334-4-32 LLC SSCS setup	80	0	4.12.3
PRIME NB OFDM PLC Physical layer counters	81	0	4.12.5
PRIME NB OFDM PLC MAC setup	82	0	4.12.6
PRIME NB OFDM PLC MAC functional parameters	83	0	4.12.7
PRIME NB OFDM PLC MAC counters	84	0	4.12.8
PRIME NB OFDM PLC MAC network administration data	85	0	4.12.9
PRIME NB OFDM PLC Application identification	86	0	4.12.11
G3-PLC MAC layer counters G3 NB OFDM PLC MAC layer counters	90	1 0	4.13.3 5.13.3
G3-PLC MAC setup G3-PLC MAC setup G3 NB OFDM PLC MAC setup	91	2 1 0	4.13.4 5.13.6 5.13.3
G3-PLC 6LoWPAN adaptation layer setup G3-PLC 6LoWPAN adaptation layer setup G3 NB OFDM PLC 6LoWPAN adaptation layer setup	92	2 1 0	4.13.5 5.13.7 5.13.4
Wi-SUN setup	95	0	4.17.1
Wi-SUN diagnostic	96	0	4.17.2
RPL diagnostic	97	0	4.17.3
MPL diagnostic	98	0	4.17.4
NTP Setup	100	0	4.9.7
ZigBee® SAS startup	101	0	4.15.2
ZigBee® SAS join	102	0	4.15.3
ZigBee® SAS APS fragmentation	103	0	4.15.4
ZigBee® network control	104	0	4.15.5
ZigBee® tunnel setup	105	0	4.15.6
Account	111	0	4.6.2
Credit	112	0	4.6.3

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Nom de la classe d'interface	class_id	version(s)	Paragraphe
Charge	113	0	4.6.4
Token gateway	115	0	4.6.5
Function control	122	0	4.4.10
Array manager	123	0	4.4.11
Communication port protection	124	0	4.4.12
SCHC-LPWAN setup	126	0	4.16.2.1
SCHC-LPWAN diagnostic	127	0	4.16.2.2
LoRaWAN setup	128	0	4.16.3.2.2
LoRaWAN diagnostic	129	0	4.16.3.2.5
ISO/IEC14908 Identification	130	0	4.18.2
ISO/IEC 14908 Protocol setup	131	0	4.18.3
ISO/IEC 14908 protocol status	132	1	4.18.4
ISO/IEC 14908 diagnostic	133	1	4.18.5
HS-PLC ISO/IEC 12139-1 MAC setup	140	0	4.14.2
HS-PLC ISO/IEC 12139-1 CPAS setup	141	0	4.14.3
HS-PLC ISO/IEC 12139-1 IP SSAS setup	142	0	4.14.4
HS-PLC ISO/IEC 12139-1 HDLC SSAS setup	143	0	4.14.5
LTE monitoring	151	1 0	4.7.9 5.7.10

4.3 Classes d'interfaces pour les paramètres et les données de mesure

4.3.1 Data (class_id = 1, version = 0)

4.3.1.1 Vue d'ensemble

Cette IC permet de modéliser diverses données, telles que les données de configuration et les paramètres. Les données sont identifiées par l'attribut *logical_name*.

Data	**0…n**	**class_id = 1, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. value	CHOICE				x + 0x08
Méthodes spécifiques	**o/f**				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.1.2 Description d'attribut

4.3.1.2.1 logical_name

Identifie l'instance de l'objet "Data". Voir Article 6 et IEC 62056-6-1:2023.

4.3.1.2.2 value

Contient les données.

Le type de données dépend de l'instanciation définie par l'attribut *logical_name* et provenant éventuellement du fabricant. Il doit être choisi de manière à éviter, avec le *logical_name*, toute ambiguïté d'interprétation. Un type de données simple ou complexe énuméré en 4.1.5 peut être utilisé, sauf en cas de limitation du choix par l'Article 6.

```
CHOICE

{
 -- types de données simples

null-data                   [0],
boolean                     [3],
bit-string                  [4],
double-long                 [5],
double-long-unsigned        [6],
octet-string                [9],
visible-string              [10],
utf8-string                 [12],
bcd                         [13],
integer                     [15],
long                        [16],
unsigned                    [17],
long-unsigned               [18],
long64                      [20],
long64-unsigned             [21],
enum                        [22],
float32                     [23],
float64                     [24],
date-time                   [25],
date                        [26],
time                        [27],
delta-integer               [28],
delta-long                  [29],
delta-double-long           [30],
delta-unsigned              [31],
delta-long-unsigned         [32],
delta-double-long-unsigned [33],

-- types de données complexes
array                       [1],
structure                   [2],
compact-array               [19]

}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.2 Register (class_id = 3, version = 0)

4.3.2.1 Vue d'ensemble

Cette IC permet de modéliser une valeur issue d'un processus ou d'un état avec l'échelle et l'unité associées. Les objets "Register" ont une notion de la nature de la valeur issue du processus ou de l'état. Elle est identifiée par l'attribut *logical_name*.

Register		0...n	class_id = 3, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. value		CHOICE				x + 0x08
3. scaler_unit	(static)	scal_unit_type				x + 0x10
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x28

4.3.2.2 Description d'attribut

4.3.2.2.1 logical_name

Identifie l'instance de l'objet "Register". Voir Article 6 et IEC 62056-6-1:2023.

4.3.2.2.2 value

Contient la valeur de processus ou d'état courant.

Le type de données de la valeur dépend de l'instanciation définie par *logical_name* et éventuellement du choix du fabricant. Il doit être choisi de manière à éviter, avec le *logical_name*, toute ambiguïté d'interprétation de la valeur.

Lorsqu'un objet "Register" est utilisé en lieu et place d'un objet "Data" (sans utiliser l'attribut *scaler_unit* ou avec scaler = 0, unit = 255), alors les types de données autorisés pour l'attribut *value* de l'IC "Data" sont autorisés.

```
CHOICE
{
        -- types de données simples

null-data                [0],
bit-string               [4],
double-long              [5],
double-long-unsigned     [6],
octet-string             [9],
visible-string           [10],
utf8-string              [12],
integer                  [15],
long                     [16],
unsigned                 [17],
long-unsigned            [18],
long64                   [20],
long64-unsigned          [21],
enum                     [22],
float32                  [23],
float64                  [24],

}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.2.2.3 scaler_unit

Fournit des informations relatives à l'unité et à l'échelle de la valeur.

```
scal_unit_type::= structure
{
scaler,
unit
}

scaler:     integer
```

Il s'agit de l'exposant (en base 10) du facteur de multiplication.

NOTE Si la valeur n'est pas numérique, l'échelle de comptage doit être mise à 0.

Voir Tableau 6 pour obtenir des exemples d'utilisation de scaler_unit.

```
unit:     enum
```

Voir Tableau 5 pour les valeurs d'unité.

4.3.2.3 Description de la méthode

4.3.2.3.1 reset (data)

Force une réinitialisation de l'objet. En appelant cette méthode, la valeur est mise à la valeur par défaut. La valeur par défaut est une constante spécifique à l'instance.

```
data::= integer (0)
```

Tableau 5 – Valeurs énumérées pour les unités physiques

unit::= enum	Unité	Grandeur	Nom de l'unité	Définition SI (commentaire)
(1)	a	temps	année	
(2)	mo	temps	mois	
(3)	wk	temps	semaine	7*24*60*60 s
(4)	d	temps	jour	24*60*60 s
(5)	h	temps	heure	60*60 s
(6)	min.	temps	minute	60 s
(7)	s	temps (t)	seconde	s
(8)	°	angle (de phase)	degré	rad*180/π
(9)	°C	température (T)	degré Celsius	K-273,15
(10)	currency	devise (locale)		
(11)	m	longueur (l)	mètre	m
(12)	m/s	vitesse (v)	mètre par seconde	m/s
(13)	m^3	volume (V) r_V, constante ou valeur impulsionnelle du compteur (volume)	mètre cube	m^3
(14)	m^3	volume corrigé [a]	mètre cube	m^3
(15)	m^3/h	débit volumique	mètre cube par heure	$m^3/(60*60s)$
(16)	m^3/h	débit volumique corrigé [a]	mètre cube par heure	$m^3/(60*60s)$

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

unit::= enum	Unité	Grandeur	Nom de l'unité	Définition SI (commentaire)
(17)	m^3/d	débit volumique		$m^3/(24*60*60s)$
(18)	m^3/d	débit volumique corrigé [a]		$m^3/(24*60*60s)$
(19)	l	volume	litre	10^{-3} m^3
(20)	kg	masse (m)	kilogramme	
(21)	N	force (F)	newton	
(22)	Nm	énergie	newton-mètre	J = Nm = Ws
(23)	Pa	pression (p)	pascal	N/m^2
(24)	bar	pression (p)	bar	10^5 N/m^2
(25)	J	énergie	joule	J = Nm = Ws
(26)	J/h	puissance thermique	joule par heure	J/(60*60s)
(27)	W	puissance active (P)	watt	W = J/s
(28)	VA	puissance apparente (S)	volt-ampère	
(29)	var	puissance réactive (Q)	var	
(30)	Wh	énergie active r_W, constante ou valeur impulsionnelle de l'énergie active du compteur	watt-heure	W*(60*60s)
(31)	VAh	énergie apparente r_S, constante ou valeur impulsionnelle de l'énergie apparente du compteur	volt-ampère-heure	VA*(60*60s)
(32)	varh	énergie réactive r_B, constante ou valeur impulsionnelle de l'énergie réactive du compteur	var-heure	var*(60*60s)
(33)	A	intensité (I)	ampère	A
(34)	C	charge électrique (Q)	coulomb	C = As
(35)	V	tension (U)	volt	V
(36)	V/m	intensité du champ électrique (E)	volt par mètre	V/m
(37)	F	capacité (C)	farad	C/V = As/V
(38)	Ω	résistance (R)	ohm	Ω = V/A
(39)	$\Omega m^2/m$	résistivité (ρ)		Ωm
(40)	Wb	flux magnétique (Φ)	weber	Wb = Vs
(41)	T	densité du flux magnétique (B)	tesla	Wb/m^2
(42)	A/m	intensité du champ magnétique (H)	ampère par mètre	A/m
(43)	H	inductance (L)	henry	H = Wb/A
(44)	Hz	fréquence (f, ω)	hertz	1/s
(45)	1/(Wh)	R_W, constante ou valeur impulsionnelle de l'énergie active du compteur		
(46)	1/(varh)	R_B, constante ou valeur impulsionnelle de l'énergie réactive du compteur		
(47)	1/(VAh)	R_S, constante ou valeur impulsionnelle de l'énergie apparente du compteur		
(48)	V^2h	r_{U2h}, constante ou valeur impulsionnelle du compteur de volt carré-heures	volt carré-heures	V^2(60*60s)
(49)	A^2h	r_{I2h}, constante ou valeur impulsionnelle du compteur d'ampère carré-heures	ampère carré-heures	A^2(60*60s)
(50)	kg/s	flux massique	kilogramme par seconde	kg/s

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

unit::= enum	Unité	Grandeur	Nom de l'unité	Définition SI (commentaire)
(51)	S, mho	conductance	siemens	1/Ω
(52)	K	température (T)	kelvin	
(53)	$1/(V^2h)$	R_{U2h}, constante ou valeur impulsionnelle du compteur de volt carré-heures		
(54)	$1/(A^2h)$	R_{I2h}, constante ou valeur impulsionnelle du compteur d'ampère carré-heures		
(55)	$1/m^3$	R_V, constante ou valeur impulsionnelle du compteur (volume)		
(56)		pourcentage	%	
(57)	Ah	ampère-heure	ampère-heure	
…				
(60)	Wh/m^3	énergie par volume	$3,6*10^3$ J/m^3	
(61)	J/m^3	valeur calorifique, wobbe		
(62)	% en mole	fraction molaire de la composition gazeuse	mole pour-cent	(unité de composition gazeuse de base)
(63)	g/m^3	masse volumique, quantité de matière		(analyse de gaz, éléments annexes)
(64)	Pa s	viscosité dynamique	pascal seconde	(caractéristique du flux gazeux)
(65)	J/kg	énergie spécifique NOTE Quantité d'énergie par unité de masse d'une substance	joule/kilogramme	$m^2 . kg . s^{-2} / kg = m^2 . s^{-2}$
….				
(70)	dBm	intensité du signal, dB milliwatt (système GSM radio, par exemple)		
(71)	dbµV	intensité du signal, dB microvolt		
(72)	dB	unité logarithmique exprimant le rapport entre deux valeurs d'une grandeur physique		
…				
(253)		réservé		
(254)	autre	autre unité		
(255)	nombre	sans unité, sans dimension		

[a] L'utilisation de ces unités (14, 16 et 18) est déconseillée, car les codes OBIS spécifient un volume corrigé (flux). Ceci permet d'éviter toute contradiction avec les unités associées aux codes OBIS sélectionnés.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 6 – Exemples d'utilisation de scaler_unit

Valeur	Scalaire	Unité	Données
263788	-3	m^3	263 788 m^3
593	3	Wh	593 kWh
3467	-1	V	346,7
3467	0	V	3467 V
3467	1	V	34 670 V

4.3.3 Extended register (class_id = 4, version = 0)

4.3.3.1 Vue d'ensemble

Cette IC permet de modéliser une valeur issue d'un processus avec ses informations associées relatives au scalaire, à l'unité, à l'état et à l'instant d'acquisition. Les objets "Extended register" ont une notion de la nature de la valeur issue du processus. Elle est décrite par l'attribut *logical_name*.

Extended register		**0...n**	**class_id = 4, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. value	(dyn.)	CHOICE				x + 0x08
3. scaler_unit	(static)	scal_unit_type				x + 0x10
4. status	(dyn.)	CHOICE				x + 0x18
5. capture_time	(dyn.)	octet-string				x + 0x20
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x38

4.3.3.2 Description d'attribut

4.3.3.2.1 logical_name

Identifie l'instance de l'objet "Extended register". Voir 6.3.1.

4.3.3.2.2 value

Voir la spécification de l'IC "Register" en 4.3.3.2.2.

4.3.3.2.3 scaler_unit

Voir la spécification de l'IC "Register" en 4.3.3.2.3.

4.3.3.2.4 status

Fournit des informations d'état spécifiques à "Extended register". La signification des éléments d'état doit être fournie pour chaque instance de l'objet.

Le type de données et le codage dépendent de l'instanciation et éventuellement du choix du fabricant. Pour l'interprétation, des informations complémentaires fournies par le fabricant peuvent être nécessaires.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
CHOICE
{
        -- types de données simples

null-data                [0],
bit-string               [4],
double-long-unsigned     [6],
octet-string             [9],
visible-string           [10],
utf8-string              [12],
unsigned                 [17],
long-unsigned            [18],
long64-unsigned          [21]
}
```

Def. Selon la définition du type d'état.

4.3.3.2.5 capture_time

Fournit une information de date et heure spécifique à "Extended register" indiquant le moment où la valeur de l'attribut *value* a été saisie.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

4.3.3.3 Description de la méthode

4.3.3.3.1 reset (data)

Cette méthode force une réinitialisation de l'objet. En appelant cette méthode, la valeur par défaut est affectée à l'attribut *value*. La valeur par défaut est une constante spécifique à l'instance.

L'attribut *capture_time* est défini à l'heure de l'exécution de la réinitialisation.

data::= integer (0)

4.3.4 Demand register (class_id = 5, version = 0)

4.3.4.1 Vue d'ensemble

Cette IC permet de modéliser une valeur de demande avec ses informations associées relatives au scalaire, à l'unité, à l'état et à l'instant d'acquisition. Un objet "Demand register" mesure et calcule périodiquement une valeur moyenne courante (*current_average_value*) et stocke une dernière valeur moyenne (*last_average_value*). L'intervalle de temps *T* sur lequel la demande est calculée est défini en spécifiant *number_of_periods* et *period*. Voir Figure 8.

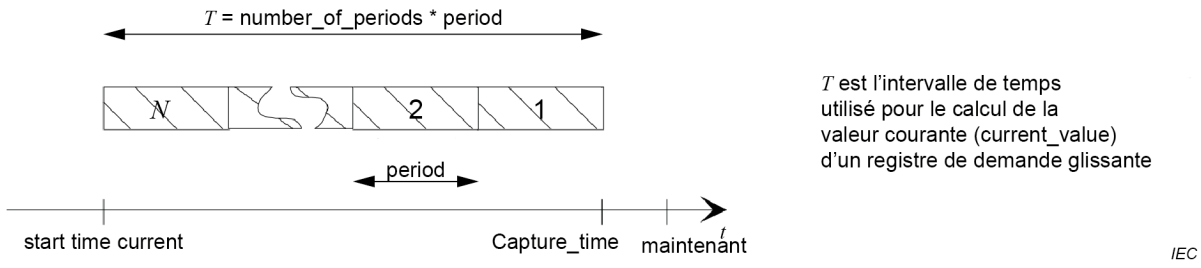


Figure 8 – Attributs de temps pour mesurer une demande glissante

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Le registre de demande délivre deux types de demandes: *current_average_value* et *last_average_value* (voir Figure 9 et Figure 10).

Les objets "Demand register" ont une notion de la nature de la valeur issue du processus qui est décrite par l'attribut *logical_name*.

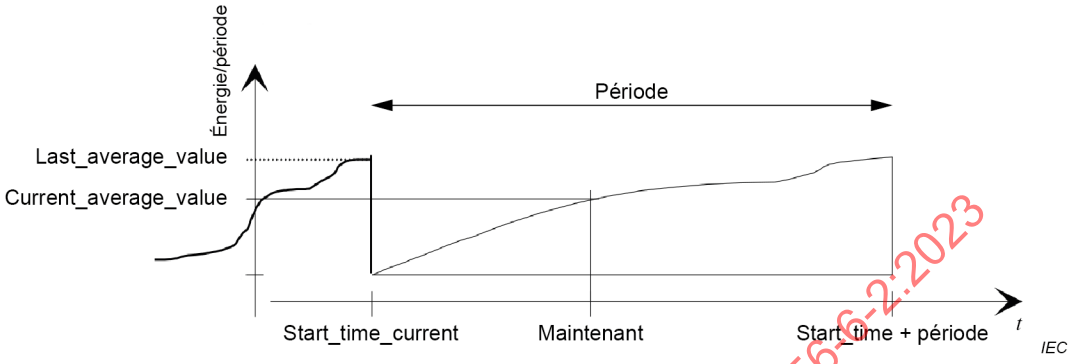
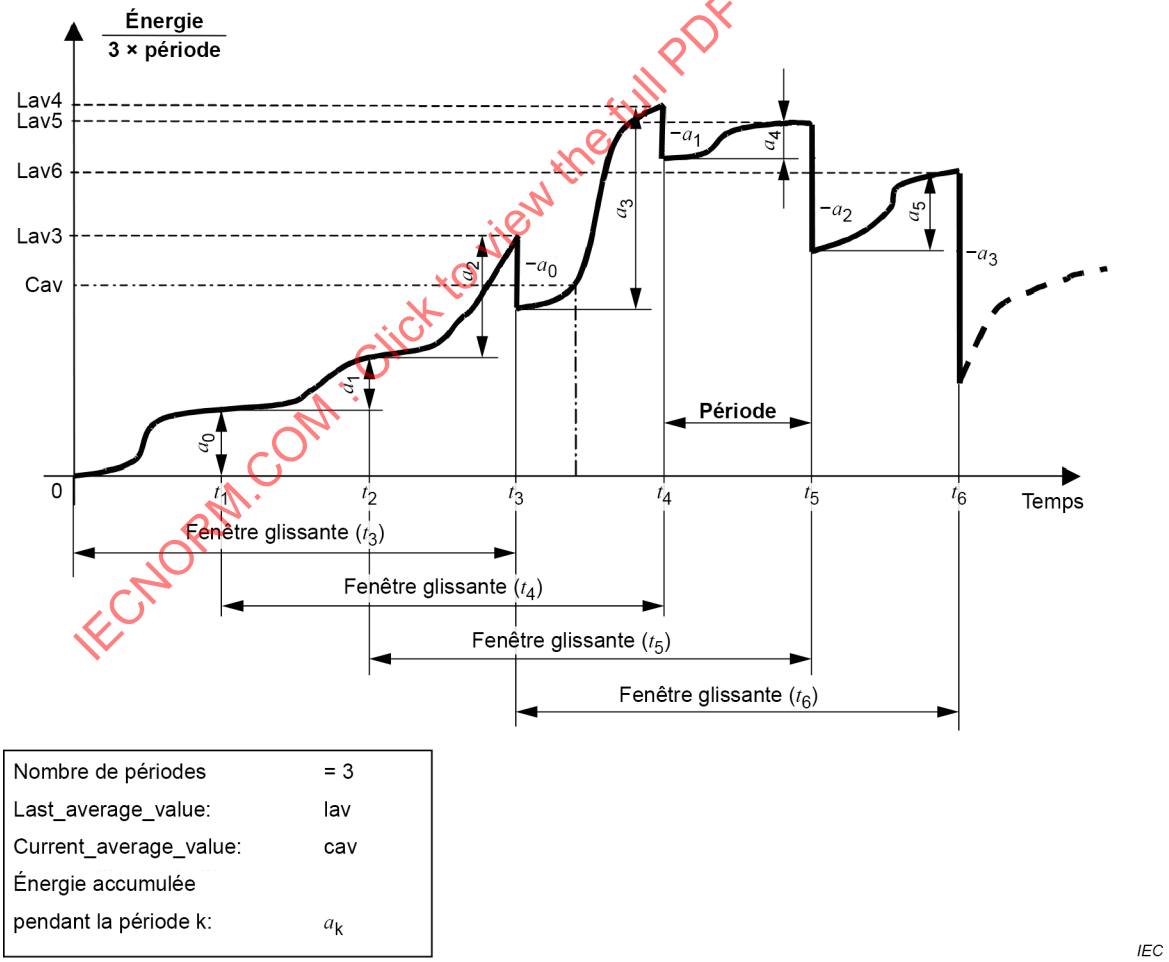


Figure 9 – Attributs dans le cas d'une demande en bloc



Nombre de périodes	= 3
Last_average_value:	lav
Current_average_value:	cav
Énergie accumulée pendant la période k:	a_k

IEC

Figure 10 – Attributs dans le cas d'une demande glissante (nombre de périodes = 3)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Demand register		**0...n**	**class_id = 5, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. current_average_value	(dyn.)	CHOICE			0	x + 0x08
3. last_average_value	(dyn.)	CHOICE			0	x + 0x10
4. scaler_unit	(static)	scal_unit_type				x + 0x18
5. status	(dyn.)	CHOICE				x + 0x20
6. capture_time	(dyn.)	octet-string				x + 0x28
7. start_time_current	(dyn.)	octet-string				x + 0x30
8. period	(static)	double-long-unsigned	1			x + 0x38
9. number_of_periods	(static)	long-unsigned	1		1	x + 0x40
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x48
2. next_period (data)		f				x + 0x50

4.3.4.2 Description d'attribut

4.3.4.2.1 logical_name

Identifie l'instance de l'objet "Demand register". Voir 6.3.1 et IEC 62056-6-1:2023.

4.3.4.2.2 current_average_value

Fournit la valeur courante (demande en cours) de l'énergie accumulée depuis *start_time*, divisée par *number_of_periods*period*.

Le type de données de la valeur dépend de l'instanciation définie par *logical_name* et éventuellement du choix du fabricant. Le type doit être choisi de manière à éviter, avec le *logical_name*, toute ambiguïté d'interprétation de la valeur.

Si une grandeur autre que l'énergie est mesurée, d'autres méthodes de calcul peuvent s'appliquer (par exemple pour calculer les valeurs moyennes de la tension ou de l'intensité).

CHOICE Pour les types de données, voir l'attribut value de l'IC "Register" (4.3.2.2.2).

4.3.4.2.3 last_average_value

Fournit la valeur de l'énergie accumulée (sur les dernières number_of_periods*period) divisée par number_of_periods*period. L'énergie de la période en cours (non achevée) n'est pas prise en compte dans le calcul.

Si une grandeur autre que l'énergie est mesurée, d'autres méthodes de calcul peuvent s'appliquer (par exemple pour calculer les valeurs moyennes de la tension ou de l'intensité).

CHOICE Pour les types de données, voir l'attribut value de l'IC "Register" (4.3.2.2.2).

4.3.4.2.4 scaler_unit

Voir la spécification de l'IC "Register" en 4.3.2.2.3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.4.2.5 status

Fournit des informations d'état spécifiques à "Demand register". Le type de données et le codage dépendent de l'instanciation et éventuellement du choix du fabricant. Pour l'interprétation, des informations complémentaires fournies par le fabricant peuvent être nécessaires.

CHOICE Pour les types de données, voir l'attribut *status* de l'IC "Extended register" (4.3.3.2.4).

Def. Selon la définition du type d'état.

4.3.4.2.6 capture_time

Fournit la date et l'heure du calcul de *last_average_value*.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

4.3.4.2.7 start_time_current

Fournit la date et l'heure du début du mesurage de *current_average_value*.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.4.2.8 period

Une période correspond à l'intervalle entre deux mises à jour successives de last_average_value. (number_of_periods*period est le dénominateur pour le calcul de la demande).

double-long-unsigned Période de mesure en secondes

Le comportement du compteur après l'écriture d'une nouvelle valeur dans cet attribut doit être spécifié par le fabricant.

4.3.4.2.9 number_of_periods

Le nombre de périodes utilisées pour calculer last_average_value. number_of_periods >= 1

– number_of_periods > 1 indique que last_average_value représente une "demande glissante",
– number_of_periods = 1 indique que last_average_value représente une "demande en bloc".

Le comportement du compteur après l'écriture d'une nouvelle valeur dans cet attribut doit être spécifié par le fabricant.

4.3.4.3 Description de la méthode

4.3.4.3.1 reset (data)

Cette méthode force une réinitialisation de l'objet. L'activation de cette méthode provoque les actions suivantes:

– il est mis fin à la période courante;
– current_average_value et last_average_value sont mises à leurs valeurs par défaut;
– capture_time et start_time_current sont mises à l'heure de l'exécution de reset (data).

data::= integer (0)

4.3.4.3.2 next_period (data)

Cette méthode est utilisée pour déclencher la fin régulière (et le redémarrage) d'une période. Clôture (met fin à) la période de mesure en cours. Met à jour capture_time et start_time et copie current_average_value dans last_average_value, met current_average_value à sa valeur par défaut. Démarre la prochaine période de mesure.

NOTE L'ancienne last_average_value (et capture_time) peut être lue pendant la durée "period". L'ancienne current_average_value n'est plus disponible à l'interface.

data::= integer (0)

4.3.5 Register activation (class_id = 6, version = 0)

4.3.5.1 Vue d'ensemble

Cette IC permet de modéliser la gestion des différentes structures de tarification. Pour chaque objet "Register activation", des groupes d'objets "Register", "Extended register" ou "Demand register", modélisant différentes sortes de grandeurs (énergie active, demande active, énergie réactive, par exemple) sont attribués. Des sous-groupes de ces registres, définis par les *masques d'activation*, définissent différentes structures de tarifs (tarifs de jour et de nuit, par exemple). L'un de ces masques d'activation, *active_mask*, définit le sous-ensemble des registres attribués à l'instance d'objet "Register activation", qui est actif. Les registres non inclus dans l'attribut *register_assignment* de tout objet "Register activation" sont toujours activés par défaut.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Register activation		0...n	class_id = 6, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. register_assignment	(static)	array				x + 0x08
3. mask_list	(static)	array				x + 0x10
4. active_mask	(dyn.)	octet-string				x+ 0x18
Méthodes spécifiques		***o/f***				
1. add_register (data)		f				x + 0x30
2. add_mask (data)		f				x + 0x38
3. delete_mask (data)		f				x + 0x40

4.3.5.2 Description d'attribut

4.3.5.2.1 logical_name

Identifie l'instance de l'objet "Register activation". Voir 6.2.11.

4.3.5.2.2 register_assignment

Spécifie une liste ordonnée d'objets COSEM attribués à l'objet "Register activation". Cette liste peut contenir différentes sortes d'objets COSEM (des instances des CI "Register", "Extended register" ou "Demand register", par exemple).

```
array        object_definition

object_definition::= structure

{
class_id:          long-unsigned,
logical_name:      octet-string
}
```

4.3.5.2.3 mask_list

Spécifie une liste de masques d'activation de registres. Chaque entrée (masque) est identifiée par son mask_name et contient un tableau d'indices renvoyant aux registres attribués au masque (le premier objet dans register_assignment est référencé par l'indice 1, le deuxième objet par l'indice 2, etc.).

```
array        register_act_mask

register_act_mask::= structure

{
mask_name:              octet-string,
index_list:             index_array
}

mask_name doit être défini de manière unique au sein de l'objet
index_array::= array     unsigned
```

4.3.5.2.4 active_mask

Définit le masque actuellement actif. Le masque est défini par son mask_name (voir mask_list, 4.3.5.2.3).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

active_mask définit les registres qui sont actuellement activés. Tous les autres registres énumérés dans register_assignment sont désactivés.

4.3.5.3 Description de la méthode

4.3.5.3.1 add_register (data)

Ajoute un registre de plus à l'attribut register_assignment. Le nouveau registre est ajouté à la fin du tableau, c'est-à-dire que le dernier registre ajouté a l'indice le plus élevé. Les indices des registres existants ne sont pas modifiés.

```
data::= structure
{
class_id:          long-unsigned,
logical_name:      octet-string
}
```

4.3.5.3.2 add_mask (data)

Ajoute un autre masque à l'attribut *mask_list*. Si un masque porte déjà le même nom, il est écrasé par le nouveau masque.

data::= register_act_mask (voir 4.3.5.3.1)

4.3.5.3.3 delete_mask (data)

Supprime un masque de l'attribut *mask_list*. Le masque est défini par son nom de masque.

data::= octet-string (mask_name)

4.3.6 Profile generic (class_id = 7, version = 1)

4.3.6.1 Vue d'ensemble

Cette IC fournit un concept généralisé permettant de stocker, de trier et d'accéder à des groupes de données ou des séries de données, appelés "objets de capture" (*capture objects*). Les objets de capture sont des attributs ou éléments appropriés d'un ou plusieurs attributs d'objets COSEM. Les objets de capture sont collectés de façon périodique ou occasionnelle.

Un profil a un tampon (*buffer*) de stockage des données capturées. Pour ne récupérer qu'une partie du tampon, une plage de valeurs ou une plage d'entrées peut être spécifiée, en demandant de récupérer toutes les entrées qui sont dans les limites de la plage spécifiée.

La liste des objets de capture définit les valeurs à stocker dans le tampon (en utilisant la capture automatique ou la méthode *capture*). La liste est définie de façon statique afin d'assurer des entrées de tampon homogènes (toutes les entrées ont la même taille et la même structure). Si la liste d'objets de capture est modifiée, le *tampon* est vidé. Si le tampon est capturé par d'autres objets "Profile generic", leur *tampon* est également vidé afin de garantir l'homogénéité de leurs entrées de *tampon*.

Le *tampon* peut être défini en étant trié par l'un des objets de *capture* (l'horloge, par exemple), ou bien les entrées sont empilées (dans l'ordre "dernier entré premier sorti"). Par exemple, il est très facile de construire un "registre de demande maximale" avec une capture de profil trié à une profondeur d'une entrée et trié par un attribut *last_average_value* de l'IC "Demand register". Il est tout aussi simple de définir un profil conservant les trois valeurs les plus élevées d'une certaine période.

La taille des données du profil est déterminée par trois paramètres:

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

a) le nombre d'entrées remplies (*entries_in_use*). Il sera égal à zéro après l'effacement du profil;

b) le nombre maximal d'entrées à conserver (*profile_entries*). Si toutes les entrées sont remplies et qu'une demande de capture () se produit, l'entrée la moins importante (selon la méthode de tri demandée) sera perdue. Ce nombre maximal d'entrées peut être spécifié. Lorsqu'il change, le tampon est alors ajusté;

c) la limite physique du tampon. Cette limite dépend souvent des objets à capturer. L'objet rejettera une tentative d'établissement du nombre maximal d'entrées à une valeur supérieure à celle physiquement possible.

Profile generic		**0...n**	**class_id = 7, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. buffer	(dyn.)	compact-array ou array				x + 0x08
3. capture_objects	(static)	array				x + 0x10
4. capture_period	(static)	double-long-unsigned				x + 0x18
5. sort_method	(static)	enum			1	x + 0x20
6. sort_object	(static)	capture_object_definition				x + 0x28
7. entries_in_use	(dyn.)	double-long-unsigned	0		0	x + 0x30
8. profile_entries	(static)	double-long-unsigned	1		1	x + 0x38
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x58
2. capture (data)		f				x + 0x60
3. *réservé des versions précédentes*		f				
4. *réservé des versions précédentes*		f				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.6.2 Description d'attribut

4.3.6.2.1 logical_name

Identifie l'instance de l'objet "Profile generic". Pour obtenir des exemples, voir 6.2.17, 6.2.19, 6.2.21, 6.2.22, 6.2.43, 6.2.46, 6.2.49, 6.2.50, 6.2.61, 6.2.62, 6.2.64, 6.2.65, 6.2.66, 6.2.67, 6.3.2, 6.3.6, etc.

4.3.6.2.2 buffer

Contient une séquence d'entrées. Chaque entrée contient les valeurs des objets capturés.

```
compact-array ou array          entrée

entry::= structure

{
CHOICE
{
        -- types de données simples
    null-data                   [0],
    boolean                     [3],
    bit-string                  [4],
    double-long                 [5],
    double-long-unsigned        [6],
    octet-string                [9],
```

```
                visible-string              [10],
                utf8-string                 [12],
                bcd                         [13],
                integer                     [15],
                long                        [16],
                unsigned                    [17],
                long-unsigned               [18],
                long64                      [20],
                long64-unsigned             [21],
                enum                        [22],
                float32                     [23],
                float64                     [24],
                date-time                   [25],
                date                        [26],
                time                        [27],
                delta-integer               [28],
                delta-long                  [29],
                delta-double-long           [30],
                delta-unsigned              [31],
                delta-long-unsigned         [32],
                delta-double-long-unsigned  [33],

                -- types de données complexes
                array                       [1],
                structure                   [2],
                compact-array               [19]
                }
}
```

Le nombre et l'ordre des éléments de la structure contenant les entrées sont les mêmes que dans la définition des *capture_objects*. Le *buffer* est rempli par des captures automatiques ou par des appels ultérieurs de la méthode *capture*. La séquence des entrées dans le tableau est ordonnée en fonction de la *sort_method* spécifiée.

Par défaut: Le *buffer* est vide après une réinitialisation.

NOTE 1 La lecture du *buffer* complet ne donne que les entrées qui sont "en utilisation".

NOTE 2 La valeur d'un objet capturé peut être remplacée par "null-data" si elle peut être récupérée sans ambiguïté à partir de la valeur précédente (par exemple pour le temps: si elle peut être calculée à partir de la valeur précédente et de *capture_period*, ou pour une valeur: si elle est égale à la valeur précédente).

L'*accès sélectif* (voir 4.1.4.17) à l'attribut *buffer* peut être disponible (facultatif). Les paramètres d'accès sélectif sont tels que définis en 4.3.6.2.9.

4.3.6.2.3 capture_objects

Spécifie la liste des objets de capture qui sont attribués à l'instance d'objet.

Sur appel de la méthode *capture (data)* ou automatiquement à des intervalles définis, les attributs sélectionnés sont copiés dans le *buffer* du profil.

```
array       capture_object_definition

capture_object_definition::= structure
{
class_id:               long-unsigned,
logical_name:           octet-string,
attribute_index:        integer,
data_index:             long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– où *attribute_index* est un pointeur vers l'attribut au sein de l'objet identifié par son *class_id* et son *logical_name*. *attribute_index 1* fait référence au 1er attribut (c'est-à-dire *logical_name*), *attribute_index 2* au 2e attribut, etc.; *attribute_index 0* fait référence à tous les attributs publics;

– où *data_index* est un pointeur sélectionnant un élément spécifique de l'attribut. Le premier élément dans l'attribut *structure* est identifié par *data_index 1*. Lorsque l'attribut n'est pas une structure, *data_index* n'a pas de signification. Si l'objet de capture est le tampon d'un profil, *data_index* identifie l'objet capturé du tampon (c'est-à-dire la colonne) du profil interne;

data_index 0: fait référence à l'attribut tout entier.

4.3.6.2.4 capture_period

>= 1: Capture automatique prise par hypothèse. Spécifie la période de capture en secondes.

0: Absence de capture automatique; la capture est déclenchée en externe ou des événements de capture se produisent de manière asynchrone.

4.3.6.2.5 sort_method

Si le profil n'est pas trié, il fonctionne comme un tampon "premier entré, premier sorti" (il est donc trié par capture, et pas nécessairement par le temps maintenu dans l'objet "Clock"). Si le tampon est saturé, l'appel de *capture* () suivant expulsera la première entrée (la plus ancienne) du tampon pour dégager de la place pour la nouvelle entrée.

Si le profil est trié, un appel de *capture* () stockera la nouvelle entrée à l'emplacement approprié dans le tampon, en déplaçant toutes les entrées suivantes et en perdant probablement l'entrée la moins intéressante. Si la nouvelle entrée entrait dans le *tampon* après la dernière entrée, et si le *tampon* est déjà plein, la nouvelle entrée ne sera pas conservée du tout.

enum:
(1) fifo (première entrée, première sortie);
(2) lifo (dernière entrée, première sortie);
(3) largest (la plus grande);
(4) smallest (la plus petite);
(5) nearest_to_zero (la plus proche de zéro);
(6) farest_from_zero (la plus éloignée de zéro).

Def. fifo

4.3.6.2.6 sort_object

Si le profil est trié, cet attribut spécifie le registre ou l'horloge servant de base à l'ordre de classement.

capture_object_definition Voir 4.3.6.2.3

Def. aucun objet par lequel trier (uniquement possible avec sort_method fifo ou lifo)

NOTE Si la sort_method est FIFO ou LIFO, tous les éléments de la capture_object_definition spécifiant l'objet de tri peuvent être zéro.

4.3.6.2.7 entries_in_use

Compte le nombre d'entrées stockées dans le tampon. Après un appel de la méthode *reset* (), le tampon ne contient pas d'entrées, et cette valeur est nulle. Sur chaque appel ultérieur de la méthode *capture* (), cette valeur est incrémentée jusqu'au nombre maximal d'entrées qui sont stockées (voir *profile_entries,* 4.3.6.2.8).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

double-long-unsigned 0…profile_entries

Def. 0

4.3.6.2.8 profile_entries

Spécifie le nombre des entrées qui doivent être conservées dans le tampon.

double-long-unsigned 1…(limité par la taille physique)

Def. 1

4.3.6.2.9 Paramètres pour l'accès sélectif à l'attribut buffer

Le Tableau 7 définit les paramètres exigés pour l'accès sélectif à l'attribut buffer.

Tableau 7 – Paramètres pour l'accès sélectif à l'attribut buffer

Sélecteur d'accès	Paramètre d'accès	Commentaire
1	range_descriptor	Seuls les éléments de tampon correspondant au range_descriptor doivent être retournés dans la réponse.
2	entry_descriptor	Seuls les éléments de tampon correspondant à l'entry_descriptor doivent être retournés dans la réponse.

range_descriptor::= structure
{ restricting_object: capture_object_definition

(Définit le capture_object limitant la plage des entrées à récupérer. Seuls les types de données simples sont autorisés.

from_value: (l'entrée à récupérer la plus ancienne ou la plus petite)

CHOICE

{ -- types de données simples
double-long [5],
double-long-unsigned [6],
octet-string [9],
visible-string [10],
utf8-string [12],
integer [15],
long [16],
unsigned [17],
long-unsigned [18],
long64 [20],
long64-unsigned [21],
float32 [23],
float64 [24],
date-time [25],
date [26],
time [27]
}

to_value: (Entrée à récupérer la plus récente ou la plus grande)

CHOICE
{comme pour "from_value" ci-dessus}

selected_values: array capture_object_definition

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(Liste de colonnes à récupérer. Si le tableau est vide (aucune entrée), toutes les données capturées sont retournées. Sinon, seules les colonnes spécifiées dans le tableau sont retournées. Le type *capture_object_definition* est spécifié ci-dessus (*capture_objects*)).
}

entry_descriptor::= structure
{
from_entry: double-long-unsigned première entrée à récupérer,
to_entry: double-long-unsigned dernière entrée à récupérer
(si to_entry == 0: l'entrée la plus élevée possible),

from_selected_value: long-unsigned indice de la première valeur à récupérer,
to_selected_value: long-unsigned indice de la dernière valeur à récupérer

(si to_selected_value == 0: selected_value la plus élevée possible)
}

NOTE 1 from_entry et to_entry identifient les lignes, from_selected_value et to_selected_value identifient les colonnes du tampon à récupérer.

NOTE 2 La numérotation des entrées et des valeurs sélectionnées commence à 1.

4.3.6.3 Description de la méthode

4.3.6.3.1 reset (data)

Vide le *tampon*. Il n'a pas d'entrées valides après cette opération. *entries_in_use* est nul après cet appel. Cet appel ne déclenche pas d'opérations supplémentaires sur les objets de capture. Précisément, il ne réinitialise pas les attributs capturés.

data::= integer (0)

4.3.6.3.2 capture (data)

Copie les valeurs des objets à capturer dans le *tampon* en lisant chaque objet de capture. En fonction de la *sort_method* et de l'état réel du tampon, elle produit une nouvelle entrée ou un remplacement pour l'entrée la moins significative. Tant que toutes les entrées n'ont pas été déjà utilisées, l'attribut *entries_in_use* sera incrémenté.

Cet appel ne déclenche pas d'opérations supplémentaires au sein des objets de capture telles que *capture* () ou *reset* ().

À noter que, s'il est nécessaire de capturer plusieurs attributs d'un objet, ces attributs doivent être définis un à un sur la liste *capture_objects*. Si attribute_index = 0, tous les attributs sont capturés.

data::= integer (0)

4.3.6.4 Comportement de l'objet après modification de certains attributs

Toute modification de l'un des *capture_objects* décrivant la structure statique du tampon appellera automatiquement un *reset* () et cet appel se propagera à tous les autres profils capturant ce profil.

En cas de tentative d'écriture dans *profile_entries* d'une valeur trop grande pour le tampon, celle-ci sera rejetée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.6.5 Restrictions

Lors de la définition des *capture_objects*, toute référence circulaire au profil doit être évitée.

4.3.6.6 Profil utilisé pour définir un sous-ensemble de valeurs de lecture préférentielle

En attribuant la valeur 1 à *profile_entries*, un objet "Profile generic" peut être utilisé pour définir un ensemble de valeurs de lecture préférentielle. Voir aussi 6.2.19. L'attribution de la valeur 1 à *capture_period* assure que les valeurs sont mises à jour toutes les secondes.

4.3.7 Utility tables (class_id = 26, version = 0)

4.3.7.1 Vue d'ensemble

Cette IC permet d'encapsuler des données en tables de l'ANSI C12.19. Chaque "table" est représentée par une instance de cette IC, identifiée par son *logical_name*.

Utility tables		**0...n**	**class_id = 26, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. table_ID	(static)	long-unsigned				x + 0x08
3. length		double-long-unsigned				x + 0x10
4. buffer		octet-string				x + 0x18
Méthodes spécifiques		***o/f***				

4.3.7.2 Description d'attribut

4.3.7.3 logical_name

Identifie l'instance de l'objet "Utility tables". Voir 6.2.40.

4.3.7.4 table_ID

Numéro de table. Ce numéro de table est tel que spécifié dans la norme ANSI et peut être soit une table standard, soit une table du fabricant.

4.3.7.5 length

Nombre d'octets dans le tampon de la table.

4.3.7.6 buffer

Contenu de la table.

L'*accès sélectif* (voir 4.1.4.17) à l'attribut *buffer* peut être disponible (facultatif). Les paramètres d'accès sélectif sont définis dans le Tableau 8.

Tableau 8 – Paramètres pour l'accès sélectif à l'attribut buffer

Sélecteur d'accès	Paramètre	Commentaire
1	offset_access	Accès à la table par décalage et comptage utilisant offset_selector pour les données de paramètre.
2	index_access	Accès à la table par id d'élément et nombre d'éléments en utilisant index_selector pour les données de paramètre.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

offset_selector::= structure

{

Offset:	double-long-unsigned	Décalage en octets au début de la zone d'accès, par rapport au début de la table.
Count:	long-unsigned	Nombre d'octets demandés ou transférés

}

index_selector::= structure

{

Index:	array long-unsigned	Séquence d'indices pour identifier des éléments dans la hiérarchie de la table.
Count:	long-unsigned	Nombre d'éléments demandés ou transférés. Les valeurs de "count" supérieures à 1 retournent autant d'éléments jusqu'à la valeur count indiquée. Une valeur nulle, lorsqu'elle est donnée dans le contexte d'une demande, se réfère au sous-arbre entier de la hiérarchie commençant au point de sélection.

}

4.3.8 Register table (class_id = 61, version = 0)

4.3.8.1 Vue d'ensemble

Cette IC permet de regrouper des entrées homogènes, des attributs identiques de plusieurs objets, qui sont tous des instances de la même IC. De plus, dans leur *logical_name* (code OBIS), la valeur dans les groupes de valeurs A à D et F est identique. Les valeurs possibles dans le groupe de valeurs E sont définies dans l'IEC 62056-6-1:2023 sous forme de tableau: l'en-tête de tableau définit la partie commune du code OBIS et chaque cellule du tableau définit une valeur possible du groupe de valeurs E. Un objet "Register table" peut capturer les attributs de tout ou partie de ces objets.

NOTE 1 Les tableaux "Mesurage d'angle de phase étendu" (voir Tableau 17) et "Creux de tension UNIPEDE" (voir Tableau 19) en sont des exemples.

NOTE 2 Si une fonctionnalité plus complexe est nécessaire, l'IC "Profile generic" peut être utilisée.

Register table		**0...n**	**class_id = 61, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. table_cell_values	(dyn.)	compact-array ou array				x + 0x08
3. table_cell_definition	(static)	structure				x + 0x10
4. scaler_unit	(static)	scaler_unit_type				x + 0x18
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x28
2. capture (data)		f				x + 0x30

4.3.8.2 Description d'attribut

4.3.8.2.1 logical_name

Identifie l'instance de l'objet "Register table".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Lorsque le format du *logical_name* est A.B.C.D.255.F, les valeurs A à D et F définissent la partie commune du nom logique des objets, dont les attributs sont capturés. Un seul attribut des objets concernés peut être capturé (par exemple l'attribut *value*).

Lorsque le format du *logical_name* est A.B.98.10.x.255, plusieurs instances de l'IC "Register table" peuvent être utilisées pour capturer différents attributs des objets concernés. Le groupe de valeurs E numérote les instances. Voir IEC 62056-6-1:2023, 6.4.

4.3.8.2.2 table_cell_values

Contient la valeur des attributs capturés, telle qu'elle serait retournée sur une GET.request ou Read.request aux attributs individuels.

```
compact-array ou array          table_cell_entry

table_cell_entry::=   CHOICE
{
      -- types de données simples
      null-data                 [0],
      bit-string                [4],
      double-long               [5],
      double-long-unsigned      [6],
      octet-string              [9],
      visible-string            [10],
      utf8-string               [12],
      bcd                       [13],
      integer                   [15],
      long                      [16],
      unsigned                  [17],
      long-unsigned             [18],
      long64                    [20],
      long64-unsigned           [21],
      float32                   [23],
      float64                   [24],

      -- types de données complexes
      structure                 [2]
}
```

Lorsque l'attribut capturé est attribute_0, les valeurs redondantes peuvent être remplacées par "null-data", si leur valeur peut être récupérée sans ambiguïté (par exemple, *scaler_unit*).

4.3.8.2.3 table_cell_definition

Spécifie la liste des attributs capturés dans le tableau de registre.

```
structure

      {
            class_id:           long-unsigned,
            logical_name:       octet-string,
            group_E_values:     array unsigned,
            attribute_index:    integer
      }
```

où:

– class_id définit le class_id commun des objets dont les attributs sont capturés;

– logical_name contient le nom logique commun des objets, avec E = 255 (caractère générique);

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– les group_E_values contiennent la liste des identifiants de cellules, de type *unsigned*, tels que définis dans le tableau correspondant de l'IEC 62056-6-1:2023;
– attribute_index est un pointeur vers l'attribut au sein de l'objet. attribute_index 0 se réfère à tous les attributs publics.

Si le *logical_name* de l'objet "Register table" est au format A.B.C.D.255.F et que les attributs définis de tous les objets identifiés dans le tableau correspondant de l'IEC 62056-6-1:2023 sont capturés, l'attribut 3 peut ne pas être accessible. Dans ce cas:

– le class_id doit être 1 "Data", 3 "Register" ou 4 "Extended register";
– le *logical_name* des objets à capturer est défini par le *logical_name* de l'objet "Register table" et le tableau correspondant de l'IEC 62056-6-1:2023.

l'indice d'attribut doit être 2 (value).

4.3.8.2.4 scaler_unit

Voir la description de l'IC "Register" en 4.3.2.2.3.

Si les attributs "value" des objets "Register" ou "Extended register" sont capturés, l'attribut *scaler_unit* doit être commun à tous les objets et cet attribut doit contenir une copie.

Si d'autres attributs ou IC sont capturés, l'attribut *scaler_unit* n'a aucune signification et doit être inaccessible.

4.3.8.3 Description de la méthode

4.3.8.3.1 reset (data)

Efface les *table_cell_values*. Il n'a pas d'effet sur les attributs capturés.

data::= integer (0)

4.3.8.3.2 capture (data)

Copie les valeurs des attributs dans les *table_cell_values*. Si attribute_index = 0, tous les attributs sont capturés.

4.3.8.4 Comportement de l'objet après modification de l'attribut table_cell_definition

Toute modification apportée à cet attribut appellera automatiquement la méthode *reset* (data) et elle sera propagée à tous les profils capturant cet objet.

En cas de tentative d'écriture dans *table_cell_definition* d'une valeur trop grande pour le tampon contenant l'attribut *table_cell_values*, elle sera rejetée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.9 Status mapping (class_id = 63, version = 0)

4.3.9.1 Vue d'ensemble

Cette IC permet de modéliser la mise en correspondance des bits dans un mot d'état avec des entrées dans un tableau de référence.

Status mapping		**0...n**	**class_id = 63, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. status_word	(dyn.)	CHOICE				x + 0x08
3. mapping_table	(static)	structure				x + 0x10
Méthodes spécifiques		***o/f***				

4.3.9.2 Description d'attribut

4.3.9.2.1 logical_name

Identifie les instances de l'objet "Status mapping". Voir 6.2.46, 6.2.49, 6.2.50, 6.2.55 et 6.3.7.

4.3.9.2.2 status_word

Contient la valeur courante du mot d'état.

```
CHOICE
{
bit-string                  [4],
double-long-unsigned        [6],
octet-string                [9],
visible-string,             [10],
utf8-string                 [12],
unsigned                    [17],
long-unsigned               [18],
long64-unsigned             [21]
}
```

La taille du *status_word* est de n*8 bits, la taille maximale étant de 65 536 bits.

Les fabricants peuvent choisir un type quelconque parmi ceux énumérés ci-dessus. Cependant, le mot d'état est toujours interprété comme une chaîne binaire (bit-string).

4.3.9.2.3 mapping_table

Contient la mise en correspondance du status_word avec les positions dans le tableau de référence.

```
structure
{
ref_table_id:               unsigned,
ref_table_mapping:          CHOICE
{
        long-unsigned               [18],
        array long-unsigned         [1]
}
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Où:

– ref_table_id identifie le tableau des états de référence;

– si "long-unsigned" est choisi, la valeur pointe vers une entrée du tableau de référence. Cette entrée est mise en correspondance avec le premier bit du mot d'état. L'entrée suivante est mise en correspondance avec le bit suivant, et ainsi de suite. La dernière entrée mise en correspondance avec le dernier bit est déterminée par la longueur du mot d'état;

– si "array" est choisi, les éléments du tableau pointent vers les entrées du tableau des états de référence. L'ordre des éléments dans le tableau correspond à la position dans le mot d'état. Le premier élément du tableau met en correspondance l'entrée de tableau référencée avec le premier bit, et le dernier élément avec le dernier bit.

4.3.10 Compact data (class_id: 62, version: 1)

4.3.10.1 Vue d'ensemble

NOTE Cette version 1 prend en charge l'accès sélectif relatif et absolu.

Les instances de l'IC "Compact data" permettent de capturer les valeurs des attributs d'objet COSEM telles que déterminées par l'attribut *capture_objects*. La capture peut avoir lieu:

- sur un déclencheur externe (capture explicite);
- à la lecture de l'attribut compact_buffer (capture implicite);

telle que déterminée par l'attribut *capture_method*.

Les valeurs sont stockées dans l'attribut *compact_buffer* sous forme de chaîne d'octets.

L'ensemble de types de données est identifié par l'attribut *template_id*. Le type de données de chaque attribut capturé est détenu par l'attribut *template_description*.

Le client peut reconstruire les données sous forme non compactée (c'est-à-dire en incluant le descripteur d'attribut COSEM, le type de données et les valeurs de données) à l'aide des attributs *capture_objects*, *template_id* et *template_description*.

Compact data		**0...n**	**class_id = 62, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. compact_buffer	(dyn.)	octet-string				x + 0x08
3. capture_objects	(static)	array				x + 0x10
4. template_id	(static)	unsigned				x + 0x18
5. template_description	(dyn.)	octet-string				x + 0x20
6. capture_method	(static)	enum				x + 0x28
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x38
2. capture (data)		f				x + 0x40

4.3.10.2 Description d'attribut

4.3.10.2.1 logical_name

Identifie l'instance de l'objet "Compact data". Voir 6.2.41.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.10.2.2 compact_buffer

Contient les valeurs des attributs capturés sous la forme d'un *octet-string*.

Si les données capturées sont de type *octet-string, bit-string, visible-string, utf8-string* ou *array*, la longueur est également incluse ici.

4.3.10.2.3 capture_objects

Spécifie la liste des attributs d'objet COSEM attribués à l'instance de l'objet "Compact data".

Lors de la définition de l'attribut *capture_object*, les références circulaires doivent être évitées.

L'attribut *template_id* doit être le premier élément du tableau *capture_objects*.

À l'appel explicite ou implicite de la méthode *capture* (data), les valeurs des attributs sélectionnés sont acquises dans le *compact_buffer*.

Deux mécanismes d'accès sélectif mutuellement exclusifs sont disponibles:

– l'accès sélectif relatif, c'est-à-dire que les entrées définies selon la date ou l'entrée en cours sont renvoyées: ce mécanisme est commandé par l'élément *data_index*;
– l'accès sélectif absolu, c'est-à-dire que les entrées situées dans une plage de dates ou d'entrées explicitement définie sont renvoyées: ce mécanisme est commandé par l'élément *restriction*.

```
array capture_object_definition

capture_object_definition::=   structure
{
class_id:                      long-unsigned,
logical_name:                  octet-string,
attribute_index:               integer,
data_index:                    long-unsigned,
restriction:                   restriction_element
}
```

Où:

– attribute_index est un pointeur vers l'attribut au sein de l'objet, identifié par class_id et logical_name; attribute_index 1 fait référence au 1er attribut (c'est-à-dire le *logical_name*), attribute_index 2 au 2e attribut, etc.; attribute_index 0 fait référence à tous les attributs publics;
– data_index est un pointeur qui sélectionne un ou plusieurs éléments particuliers d'un attribut avec un type de données complexe (structure ou tableau):
  - si le type de données de l'attribut est simple, alors data_index n'a aucune signification;
  - si le type de données de l'attribut est une structure ou un tableau, data_index pointe vers un ou plusieurs éléments spécifiques de la structure ou du tableau;
  - lorsque l'attribut est le *buffer* d'un objet "Profile generic", data_index contient des paramètres d'accès sélectif en fonction de la date ou de l'entrée en cours.

data_index:	**MS-Byte**		**LS-Byte**
	Quartet supérieur	Quartet inférieur	

  - 0x0000 = identifie tout l'attribut;
  - 0x0001 à 0x0FFF = identifie un élément dans l'attribut complexe. Le premier élément de l'attribut complexe est identifié par data_index 1;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

- 0x1000 à 0xFFFF = accès sélectif au tableau contenant le *buffer* d'un objet "Profile generic". Le data-index sélectionne les entrées sur un certain nombre de périodes (récentes) ou un certain nombre d'entrées (récentes), ainsi que les colonnes du tableau.

Le codage est spécifié dans le Tableau 21.

Lorsque l'attribut est le *buffer* d'un objet "Profile generic", restriction_element spécifie des paramètres d'accès sélectif dans une plage de dates ou d'entrées définie de manière explicite.

```
restriction_element::=   structure
{
restriction_type:  enum:
                        (0) aucune,
                        (1) restriction par date,
                        (2) restriction par entrée
restriction_value: CHOICE
{
      null-data,                  // pas de restriction
      restriction_by_date,
      restriction_by_entry
}
}

restriction_by_date::=   structure
{
from_date:        octet-string,
to_date:          octet-string
}

restriction_by_entry::=  structure
{
from_entry:       double-long-unsigned,
to_entry:         double-long-unsigned
}
```

– restriction_element définit l'accès sélectif absolu au *buffer* "Profile generic" par plage de dates (from_date à to_date) ou par entrées (from_entry à to_entry). Pour utiliser ce mécanisme d'accès sélectif absolu, data_index doit être positionné à 0x0000;

restriction_element est composé de restriction_type et de restriction_value:

- pour la restriction par plage de dates, l'élément restriction_type contient (1) (restriction par date) et l'élément restriction_value contient la structure restriction_by_date;
- pour la restriction par entrées, l'élément restriction_type contient (2) (restriction par entrée) et l'élément restriction_value contient la structure restriction_by_entry;

sinon, l'élément restriction_type contient (0) (aucune) et l'élément restriction_value contient null-data. Ce choix doit également être effectué si l'accès sélectif relatif doit être utilisé.

4.3.10.2.4 template_id

Contient l'identifiant du modèle.

Il doit identifier de manière unique l'instance de l'IC "Compact data" et le *template_description*.

4.3.10.2.5 template_description

Fournit le type de données de chaque attribut capturé. Il s'agit d'un *octet-string* généré automatiquement par le serveur lors de la programmation des *capture_objects*. Sa structure est la suivante:

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– le premier octet est 0x02 (l'étiquette d'une structure);
– il est suivi du nombre d'éléments dans la structure (le même que dans le tableau *capture_objects*) codé sous la forme d'un entier de longueur variable;
– vient ensuite le type de données de chaque attribut, dans le même ordre que dans le tableau *capture_object*:
  - dans le cas des attributs avec un type de données simple, le type de données est représenté par un seul octet, contenant l'étiquette du type de données. Dans le cas de *bit-string* [4], *octet-string* [9], *visible-string* [10], *utf8-string* [12], la longueur de la chaîne fait partie des données contenues dans *compact_buffer*;
  - dans le cas d'un *array* [1], le type de données est représenté par un seul octet 0x01. Vient ensuite la description du type des éléments du tableau. Le nombre d'éléments du tableau fait partie des données contenues dans *compact_buffer*;

dans le cas d'une *structure* [2], le type de données est représenté par un seul octet 0x02, suivi du nombre d'éléments à l'intérieur de la structure, puis de l'étiquette de chacun d'eux.

4.3.10.2.6 capture_method

Définit la manière de mettre à jour *compact_buffer*.

enum

(0) Capture à l'appel de la méthode *capture* (data). Cela peut se produire à distance ou en local (capture explicite),
(1) Capture à la lecture de l'attribut *compact_buffer* (capture implicite).

4.3.10.3 Description de la méthode

4.3.10.3.1 reset (data)

Efface le *compact_buffer*. Après l'appel de cette méthode, le tampon contient une chaîne d'octets de longueur 0, jusqu'à la nouvelle capture.

Cet appel ne déclenche pas d'opérations supplémentaires des objets de capture. Précisément, il ne réinitialise pas les attributs capturés.

data::= integer(0)

4.3.10.3.2 capture (data)

Copie les valeurs des attributs dans le *compact_buffer* en lisant chaque objet de capture.

Cet appel ne déclenche pas d'opérations supplémentaires au sein des objets de capture telles que capture () ou reset ().

data::= integer(0)

4.3.10.4 Comportement de l'objet après modification de certains attributs

Toute modification de *capture_objects* doit réinitialiser le *compact_buffer* et mettre automatiquement à jour le *template_description*.

4.3.10.5 Restrictions

Lors de la définition de l'attribut *capture_object*, les références circulaires doivent être évitées.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.10.6 Exemples d'utilisation de données compactées

4.3.10.6.1 Données de facturation quotidiennes

Le Tableau 9 présente les données de facturation quotidiennes qui sont capturées (avec le *template_id* obligatoire) dans l'attribut *compact_buffer* d'un objet "Compact data".

Tableau 9 – Données de facturation quotidiennes

Data	class_id	Nom logique	attribute_id	data_ index	Taille (octets)	Type	Valeur
1	2	3	4	5	6	7	8
ID du modèle	62	0-0:66.0.0.255	4	0	1	unsigned	0
Temps Unix	1	0-0:1.1.0.255	2	0	4	double-long-unsigned	1374573317
État de fonctionnement	1	0-0:96.5.0.255	2	0	1	unsigned	0x29
Registre d'erreurs	1	0-0:97.97.0.255	2	0	1	unsigned	0x18
Indice total	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	6422483
Indice F1	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	865234
Indice F2	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	1234567
Indice F3	3	7-0:13.83.1.255	2	0	4	double-long-unsigned	2345678
Nom du calendrier d'activités	20	0-0:13.0.0.255	2	0	6	octet-string	"ABCDEF"
Compteur d'événements	1	0-0:96.15.1.255	2	0	2	long-unsigned	7890

Le Tableau 10 présente les attributs de l'objet "Compact data".

Tableau 10 – Attributs de l'objet "Compact data"

capture_objects (array)	Pour les éléments du tableau, voir colonnes 2, 3, 4 et 5 du Tableau 9.
template_id (unsigned)	0
template_description (octet-string)	-- Pour les types de données, voir colonne 7 du Tableau 9. 02 0A 11 06 11 11 06 06 06 06 09 12
compact_buffer (octet-string) 32 octets	-- Pour les valeurs, voir colonne 8 du Tableau 9. 00 51EE5305 29 18 0061FFD3 000D33D2 0012D687 0023CACE 06414243444546 1ED2

Pour comparaison, le codage A-XDR des mêmes données, comme si elles étaient accessibles à l'aide d'un service GET-WITH-LIST, est présenté dans le Tableau 11. Seul le codage du résultat (SEQUENCE OF Get-Data-Result) est présenté.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 11 – Codage A-XDR des données (SEQUENCE DE Get-Data-Result)

Codage	Signification	Longueur
09	**SEQUENCE de 9 éléments**	1
00 06 51EE5305	double-long-unsigned	6
00 11 29	unsigned	3
00 11 18	unsigned	3
00 06 0061FFD3	double-long-unsigned	6
00 06 000D33D2	double-long-unsigned	6
00 06 0012D687	double-long-unsigned	6
00 06 0023CACE	double-long-unsigned	6
00 09 06 414243444546	octet-string de longueur 6	9
00 12 1ED2	long-unsigned	4
	Total	50 octets
NOTE Les premiers 00 de chaque élément indiquent le CHOIX "Data" dans Get-Data-Result.		

4.3.10.6.2 Données de diagnostic et d'alarme

Le Tableau 12 présente les données de diagnostic et d'alarme qui sont capturées (avec le *template_id* obligatoire) dans l'attribut *compact_buffer* d'un objet "Compact data".

Tableau 12 – Données de diagnostic et d'alarme

Data	class _id	Nom logique	attribute _id	data_ index	Taille (octets)	Type	Valeur
1	2	3	4	5	6	7	8
ID du modèle	62	0-0:66.0.0.255	4	0	1	unsigned	1
Diagnostic en cours	3	7-0:96.5.1.255	2	0	2	long-unsigned	0x4200
Diagnostic quotidien	3	7-1:96.5.1.255	2	0	2	long-unsigned	0x4108
Diagnostic de période de facturation	1	7-2:96.5.1.255	2	0	2	long-unsigned	0x4308
Compteur d'événements de synchronisation	1	0-0:96.15.2.255	2	0	2	long-unsigned	763
Version du progiciel métrologique	1	7-0:0.2.1.255	2	0	8	octet-string	"ABCDEFGH"
Compteur d'événements métrologiques	1	0-0:96.15.1.255	2	0	2	long-unsigned	1532
Version du progiciel non métrologique	1	7-1:0.2.1.255	2	0	8	octet-string	"DEFGHIJK"

Le Tableau 13 présente les attributs de l'objet "Compact data".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 13 – Attributs de l'objet "Compact data"

capture_objects (array)	Pour les éléments du tableau, voir colonnes 2, 3, 4 et 5 du Tableau 12.
template_id (unsigned)	1
template_description (octet-string)	-- Pour les types de données, voir colonne 7 du Tableau 12. 02 08 11 12 12 12 12 09 12 09
compact_buffer (octet-string) 29 octets	-- Pour les valeurs, voir colonne 8 du Tableau 12. 01 4200 4108 4308 02FB 08414243444546 4748 05FC 084445464748494A4B

Pour comparaison, le codage A-XDR des données, comme si elles étaient lues dans l'attribut *buffer* d'un objet "Profile generic", est présenté dans le Tableau 14 (seules les données sont présentées).

Tableau 14 – Codage des données lues dans l'attribut buffer d'un objet "Profile generic"

Codage	Signification	Longueur
01 01	tableau d'un élément	2
02 07	structure de 7 éléments	2
12 4200	long-unsigned	3
12 4108	long-unsigned	3
12 4308	long-unsigned	3
12 02FB	long-unsigned	3
09 08 4142434445464748	octet-string de longueur 8	10
12 05FC	long-unsigned	3
09 08 4445464748494A4B	octet-string de longueur 8	10
	Total	39 octets

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.3.10.6.3 Lecture du journal de bord

Dans cet exemple, les données à compacter sont l'attribut *buffer* d'un journal de bord détenu par l'objet "Profile generic" capturant 2 éléments (voir Tableau 15).

Tableau 15 – Données du journal de bord

Data	class_id	Nom logique	Attribute_id	data_ index	Taille (octets)	Type	Valeur
1	2	3	4	5	6	7	8
Temps Unix	1	0-0:1.1.0.255	2	0	4	double-long-unsigned	Voir Note
Code d'événement	1	0-0:96.11.2.255	2	0	1	unsigned	
NOTE Pour cet exemple, les valeurs suivantes sont prises par hypothèse: – horodatage UNIX: 1374573317D; – état: 0x29; – pour simplifier l'exemple, les valeurs sont les mêmes pour toutes les entrées; – il y a 50 entrées dans le tampon.							

Le Tableau 16 présente les données à capturer par l'objet "Compact data".

Tableau 16 – Attributs de l'objet "Compact data"

Data	class_ id	Nom logique	attribute_ id	data_ index	Taille (octets)	Type	Valeur
1	2	3	4	5	6	7	8
ID du modèle	62	0-0:66.0.0.255	4	0	1	unsigned	2
Tampon du journal de bord[1]	7	0-0:99.98.0.255	2	0	dyn. [2]	tableau de la structure	2

[1] Voir Tableau 15.

[2] La taille est dynamique et dépend du nombre d'entrées capturées.

Le Tableau 17 présente les attributs de l'objet "Compact data".

Tableau 17 – Attributs de l'objet "Compact data"

capture_objects (array)	Pour les éléments du tableau, voir colonnes 2, 3, 4 et 5 du Tableau 16. L'objet de capture ne contient que 2 éléments: le template_id et l'attribut *buffer* du journal de bord.
template_id (unsigned)	2
template-description (octet-string)	-- Pour les types de données, voir colonne 7 du Tableau 16. 02 02 11 01 02 02 06 11 Signification: 02 02 – structure de 2 éléments 11 – le premier élément est un unsigned 01 02 02– le deuxième élément est un tableau de la structure, laquelle contient deux éléments 06 – le premier est un double-long-unsigned 11 – le deuxième est un unsigned
compact_buffer (octet-string)	-- Pour les valeurs, voir colonne 8 du Tableau 16. 02 – valeur du template-id 32 – nombre d'éléments dans le tableau et 50*5 = 250 octets (pour les 50 éléments du journal de bord) 252 octets au total

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Pour comparaison, le codage A-XDR des mêmes données lues dans l'attribut *buffer* d'un objet "Profile generic", est présenté dans le Tableau 18.

Tableau 18 – Codage A-XDR des données lues dans l'attribut *buffer*

Codage	Signification	Longueur
01 32	tableau de 50 éléments	2
02 02	structure de 2 éléments	2
06 51EE5305	double-long-unsigned	5
11 29	unsigned	2
02 02	structure de 2 éléments	2
06 51EE5305	double-long-unsigned	5
11 29	unsigned	2
02 02	structure de 2 éléments	2
06 51EE5305	double-long-unsigned	5
11 29	unsigned	2
….	…	…
	Total	**452 octets**

4.4 Classes d'interfaces pour le contrôle et la gestion des accès

4.4.1 Vue d'ensemble

Les classes d'interfaces de cette catégorie modélisent la structure logique du serveur DLMS®/COSEM, ce qui permet de configurer et de gérer l'accès à ses ressources, de mettre à jour le progiciel et de gérer la sécurité:

- les classes "Association SN" (voir 4.4.3) et "Association LN" (voir 4.4.4) modélisent les AA. Leurs instances, les objets "Association", fournissent la liste des objets accessibles dans chaque AA, gèrent et contrôlent les droits d'accès à leurs attributs et méthodes. Ils gèrent également l'authentification des partenaires de communication;
- la classe "SAP Assignment" (voir 4.4.5) modélise la structure logique du serveur;
- la classe "Image transfer" (voir 4.4.6) modélise le processus de mise à jour du progiciel;
- la classe "Security setup" (voir 4.4.7) modélise les éléments du contexte de sécurité. Les objets "Security setup" sont référencés à partir des objets "Association" et permettent de configurer les suites et politiques de sécurité, et de gérer les informations de sécurité;
- la classe "Push setup" (voir 4.4.8) modélise l'opération Push du serveur;
- la classe "Data protection" (voir 4.4.9) spécifie les éléments nécessaires à l'application de la protection cryptographique des valeurs d'attribut d'objet COSEM, ainsi que les paramètres d'appel et de retour de la méthode;
- la classe "Function control" (voir 4.4.10) permet d'activer et de désactiver des fonctions dans le serveur;
- la classe "Array manager" (voir 4.4.11) permet de gérer les attributs de type *array* d'autres objets d'interface;
- la classe "Communication port protection" (voir 4.4.12) permet la protection des ports contre les tentatives de communication non autorisées.

4.4.2 Identification des utilisateurs du client

Cette nouvelle fonction permet au serveur de distinguer les différents utilisateurs côté client et de journaliser leurs activités d'accès au compteur.

Chaque AA établie entre un client et un serveur peut être utilisée par plusieurs utilisateurs côté client. Les propriétés de l'AA sont configurées dans le serveur à l'aide des objets "Association" et "Security setup". Tous les utilisateurs d'une AA côté client utilisent ces mêmes propriétés.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Les clés de sécurité sont connues du client et du serveur, mais elles n'ont pas besoin d'être connues des utilisateurs du client.

La liste des utilisateurs (identifiés par leurs user_id et user_name) est connue du client et du serveur. Elle est gérée dans le serveur par l'attribut *user_list* des objets "Association".

NOTE La manière dont un client authentifie un utilisateur qui se connecte au système client n'entre pas dans le domaine d'application de la présente spécification.

Lors de l'établissement de l'AA, l'user_id (appartenant au user_name) se trouve dans le champ calling-AE-invocation-id de l'AARQ APDU. Si l'user_id fourni se trouve dans l'*user_list*, l'AA peut être établie (à condition que toutes les autres conditions soient satisfaites) et l'attribut *current_user* est mis à jour. La valeur de cet attribut peut être journalisée.

Si le serveur ne "connaît" pas l'utilisateur, l'AA ne doit pas être établie. Le serveur peut rejeter en silence la requête d'établissement de l'AA ou renvoyer un message d'erreur approprié.

Le processus d'identification des utilisateurs est facultatif: si l'*user_list* est vide (c'est-à-dire un tableau de 0 élément), la fonction est désactivée.

4.4.3 Association SN (class_id = 12, version = 4)

4.4.3.1 Vue d'ensemble

Les dispositifs logiques COSEM capables d'établir des AA au sein d'un contexte COSEM en utilisant le référencement SN, modélisent les AA à l'aide des instances de l'IC "Association SN". Un dispositif logique COSEM peut avoir une instance de cette IC pour chaque AA que le dispositif est capable de prendre en charge.

Le **short_name** de l'objet "Association SN" lui-même est fixé dans le contexte COSEM. Voir 4.1.3.

Association SN	**0...n**	**class_id = 12, version = 4**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. object_list (static)	objlist_type				x + 0x08
3. access_rights_list (static)	access_rights_type				x + 0x10
4. security_setup_reference (static)	octet-string				x + 0x18
5. user_list (static)	array				x + 0x20
6. current_user	structure				x + 0x28
Méthodes spécifiques	***o/f***				
1. *réservé des versions précédentes*	f				
2. *réservé des versions précédentes*	f				
3. read_by_logicalname (data)	f				x + 0x30
4. *réservé des versions précédentes*	f				
5. change_secret (data)	f				x + 0x40
6. *réservé des versions précédentes*	f				
7. *réservé des versions précédentes*					
8. reply_to_HLS_authentication (data)	f				x + 0x58
9. add_user (data)	f				x + 0x60
10. remove_user (data)	f				x + 0x68

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.3.2 Description d'attribut

4.4.3.2.1 logical_name

Identifie l'instance de l'objet "Association SN". Voir 6.2.33.

4.4.3.2.2 object_list

Contient la liste de tous les objets avec leurs base_name (short_name), class_id, version et *logical_name*. Le base_name est l'objectName DLMS® du premier attribut (*logical_name*).

```
objlist_type::= array          objlist_element

objlist_element::= structure
{
base_name:      long,
class_id:       long-unsigned,
version:        unsigned,
logical_name:   octet-string
}
```

L'*accès sélectif* (voir 4.1.4.17) à l'attribut *object_list* peut être disponible. Les valeurs de sélecteur d'accès et leurs paramètres sont tels que définis dans le Tableau 19.

4.4.3.2.3 access_rights_list

Contient les droits d'accès aux attributs et méthodes.

Le lien entre l'*object_list* et l'*access_rights_list* est le base_name, présent à la fois dans la structure objlist_element et la structure access_right_element. Par conséquent, les base_names des deux listes doivent être identiques. Le nombre (et de préférence l'ordre) des éléments doit également être le même dans le tableau d'objlist_element et dans le tableau d'access_right_element.

```
access_rights_type::= array          access_rights_element

    access_rights_element::= structure
    {
    base_name:              long,
    attribute_access:       attribute_access_descriptor,
    method_access:          method_access_descriptor
    }

attribute_access_descriptor::= array      attribute_access_item

    attribute_access_item::= structure
    {
    attribute_id:           integer,
    access_mode:            enum,
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si la valeur enum est interprétée comme un unsigned8, la signification de chaque bit est la suivante:

```
                {
                    Bit     attribute access_mode

                    (0)     accès en lecture,
                    (1)     accès en écriture,
                    (2)     demande authentifiée,
                    (3)     demande cryptée,
                    (4)     demande signée numériquement,
                    (5)     réponse authentifiée,
                    (6)     réponse cryptée,
                    (7)     réponse signée numériquement,
                }


    access_selectors:       CHOICE
                    {
                    null-data           [0],
                    array integer       [1]
                    }
                }
method_access_descriptor::= array       method_access_item

                method_access_item::= structure
                {
                method_id:              integer,
                access_mode:            enum
```

Si la valeur enum est interprétée comme un unsigned8, la signification de chaque bit est la suivante:

```
                        {
                            Bit     method access_mode

                            (0)     accès,
                            (1)     non utilisé,
                            (2)     demande authentifiée,
                            (3)     demande cryptée,
                            (4)     demande signée numériquement,
                            (5)     réponse authentifiée,
                            (6)     réponse cryptée,
                            (7)     réponse signée numériquement,
                        }
                }
```

L'*accès sélectif* (voir 4.1.4.17) à l'attribut access_rights_list peut être disponible (facultatif). Les valeurs de sélecteur d'accès et leurs paramètres sont tels que définis dans le Tableau 19.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 19 – Paramètres pour l'accès sélectif aux attributs object_list et access_rights_list

Valeurs de sélecteur d'accès	Paramètre	Disponible avec l'attribut	Commentaire
1	class_id: long-unsigned	2	Délivre le sous-ensemble de l'object_list pour une class_id spécifique. Pour la réponse: data::= objlist_type

2	structure
	{
	class_id: long-unsigned,
	logical_name: octet-string
	}
		2	Délivre l'entrée de l'object_list pour un class_id et un *logical_name* spécifiques.
			Pour la réponse: data::= objlist_element

3	base_name: long	2, 3	Dans le cas de l'attribut 2, délivre l'entrée de l'object_list pour un base_name spécifique. Pour la réponse: data::= objlist_element Dans le cas de l'attribut 3, délivre l'entrée de l'access_rights_list pour un base_name spécifique. Pour la réponse: data::=access_rights_element

4.4.3.2.4 security_setup_reference

Référence l'objet "Security setup" par son *logical_name*. L'objet référencé gère la sécurité pour une instance donnée de l'objet "Association SN".

4.4.3.2.5 user_list

Contient la liste des utilisateurs autorisés à utiliser l'AA gérée par l'instance donnée de l'IC "Association SN".

```
array          user_list_entry

user_list_entry::=        structure
{
user_id:                  unsigned,

user_name                 CHOICE
                               {
                               visible-string      [10],
                               UTF-8-string        [12]
                               }
}
```

Où:

– user_id est l'identifiant de l'utilisateur (cette valeur se trouve dans le champ calling-AE-invocation-id de l'AARQ);

– user_name est le nom de l'utilisateur.

Lorsque l'attribut user_list est vide (c'est-à-dire un tableau de 0 élément), tous les utilisateurs peuvent utiliser l'AA, c'est-à-dire que le champ calling-AE-invocation-id de l'AARQ est ignoré.

Lorsque l'attribut user_list n'est pas vide, seuls les utilisateurs présents dans la liste peuvent établir l'AA, c'est-à-dire que le champ calling-AE-invocation-id de l'AARQ doit être présent et que sa valeur doit correspondre à l'un des user_ids de l'user_list, sinon l'AA n'est pas établie.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.3.2.6 current_user

Contient l'identifiant de l'utilisateur en cours.

current_user::= user_list_entry (voir 4.4.3.2.5)

Si user_list est vide, current_user doit être une structure {user_id: unsigned 0, user_name: chaîne visible de 0 élément}.

4.4.3.3 Description de la méthode

4.4.3.3.1 read_by_logicalname (data)

Lit les attributs pour les objets sélectionnés. Les objets sont spécifiés par leur class_id et leur *logical_name*. Avec cette méthode, la caractéristique d'accès paramétré peut également être utilisée.

```
data::= array       attribute_identification

attribute_identification::=      structure
{
class_id:                  long-unsigned,
logical_name:              octet-string,
attribute_index:           integer
}
```

où attribute_index est un pointeur (c'est-à-dire le décalage) vers l'attribut au sein de l'objet.

attribute_index 0 délivre tous les attributs; attribute_index 1 délivre le premier attribut (c'est-à-dire *logical_name*), etc.

Pour la réponse: les données sont fonction du type de l'attribut.

NOTE Si au moins un attribut n'a pas de droit d'accès en lecture dans le cadre de l'association courante, un *read_by_logicalname* () à l'indice d'attribut 0 révèle le message d'erreur "scope-of-access-violated" (voir IEC 62056-5-3:2023, Article 8).

4.4.3.3.2 change_secret (data)

Change le secret LLS ou HLS (le mot de passe, par exemple).

```
data::= octet-string       new secret
```

NOTE 1 La structure de "new secret" dépend du mécanisme de sécurité mis en œuvre. Le "new secret" peut contenir des bits de contrôle supplémentaires et il peut être crypté.

NOTE 2 Dans le cas de HLS avec GMAC, le (HLS_)secret est détenu par l'objet "Security setup" référencé dans l'attribut.

4.4.3.3.3 reply_to_HLS_authentication (data)

L'appel à distance de cette méthode délivre au serveur le résultat du traitement de secret par le client du défi lancé par le serveur au client, f(StoC), comme le paramètre de service *data* de la primitive Read.request appelée avec l'accès paramétré.

```
data::= octet-string       réponse du client au défi
```

Si l'authentification est acceptée, la réponse (primitive Read.confirm) contient Result == OK et le résultat du traitement de secret par le serveur du défi du client au serveur, f(CtoS) dans le paramètre de service *data* du service Read.response.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

data::= octet-string réponse du serveur au défi

Si l'authentification n'est pas acceptée, le paramètre *result* dans la réponse doit contenir une valeur non-OK et aucune donnée ne doit être renvoyée.

4.4.3.3.4 add_user (data)

Ajoute un utilisateur à l'user_list.

data::= user_list_entry (voir 4.4.3.2.5)

4.4.3.3.5 remove_user (data)

Supprime un utilisateur de l'user_list.

data::= user_list_entry (voir 4.4.3.2.5)

4.4.4 Association LN (class_id = 15, version = 3)

4.4.4.1 Vue d'ensemble

Les dispositifs logiques COSEM capables d'établir des AA au sein d'un contexte COSEM en utilisant le référencement LN, modélisent les AA par des instances de l'IC "Association LN". Un dispositif logique COSEM dispose d'une instance de cette IC pour chaque AA que le dispositif est capable de prendre en charge.

Association LN		**0...MaxNbofAss.**	**class_id = 15, version = 3**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. object_list	(static)	object_list_type				x + 0x08
3. associated_partners_id		associated_partners_type				x + 0x10
4. application_context_name		context_name_type				x + 0x18
5. xDLMS_context_info		xDLMS_context_type				x + 0x20
6. authentication_mechanism_name		mechanism_name_type				x + 0x28
7. secret		octet-string				x + 0x30
8. association_status		enum				x + 0x38
9. security_setup_reference	(static)	octet-string				x + 0x40
10. user_list	(static)	array				x + 0x48
11. current_user		structure				x + 0x50
Méthodes spécifiques		***o/f***				
1. reply_to_HLS_authentication (data)		f				x + 0x60
2. change_HLS_secret (data)		f				x + 0x68
3. add_object (data)		f				x + 0x70
4. remove_object (data)		f				x + 0x78
5. add_user (data)		f				x + 0x80
6. remove_user (data)		f				x + 0x88

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2 Description d'attribut

4.4.4.2.1 logical_name

Identifie l'instance de l'objet "Association LN". Voir 6.2.33.

4.4.4.2.2 object_list

4.4.4.2.2.1 Généralités

Contient la liste des objets COSEM visibles avec leurs class_id, version, *logical_ name* et les droits d'accès à leurs attributs et méthodes au sein de l'AA donnée.

```
object_list_type::= array              object_list_element
object_list_element::= structure

{
class_id:              long-unsigned,
version:               unsigned,
logical_name:          octet-string,
access_rights:         access_right
}

access_right ::= structure
{
attribute_access:      attribute_access_descriptor,
method_access:         method_access_descriptor
}

attribute_access_descriptor::=array     attribute_access_item

attribute_access_item::=    structure
{
attribute_id:          integer,
access_mode:           enum
```

Si la valeur enum est interprétée comme un unsigned8, la signification de chaque bit est la suivante:

```
{
      Bit    attribute access_mode

      (0)    accès en lecture,
      (1)    accès en écriture,
      (2)    demande authentifiée,
      (3)    demande cryptée,
      (4)    demande signée numériquement,
      (5)    réponse authentifiée,
      (6)    réponse cryptée,
      (7)    réponse signée numériquement,
}

access_selectors:      CHOICE
{
      null-data        [0],
      array integer    [1]
}
}

method_access_descriptor::= arraymethod_access_item

method_access_item::= structure
{
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

method_id: integer,
access_mode: enum

Si la valeur enum est interprétée comme un unsigned8, la signification de chaque bit est la suivante:
{
Bit method access_mode

(0) accès,
(1) non utilisé,
(2) demande authentifiée,
(3) demande cryptée,
(4) demande signée numériquement,
(5) réponse authentifiée,
(6) réponse cryptée,
(7) réponse signée numériquement,
}
}

Où:

- les éléments attribute_access_descriptor et method_access_descriptor contiennent toujours tous les attributs ou méthodes mis en œuvre;
- les access_selectors contiennent une liste des valeurs de sélecteur prises en charge.

L'*accès sélectif* (voir 4.1.4.17) à l'attribut *object_list* peut être disponible (facultatif). Les paramètres d'accès sélectif sont tels que définis en 4.4.4.2.2.2.

4.4.4.2.2.2 Paramètres pour l'accès sélectif à l'attribut *object_list*

- Si aucun accès sélectif n'est demandé (aucun paramètre Access_Selection_Parameters n'est présent dans la primitive de service GET.request (.indication) pour l'attribut object_list), le service.response (.confirmation) correspondant doit contenir tous les object_list_elements de l'attribut object_list.
- Lorsqu'un accès sélectif est demandé à l'attribut object_list (le paramètre Access_Selection_Parameter est présent), la réponse doit contenir une liste "filtrée" d'object_list_elements, comme indiqué dans le Tableau 20:

Tableau 20 – Paramètres pour l'accès sélectif à l'attribut object_list

Sélecteur d'accès	Paramètre d'accès	Commentaire
1	NULL	Toutes les informations, à l'exclusion des access_rights, doivent être incluses dans la réponse.
2	class_list	Accès par class_id. Dans ce cas, seuls doivent être inclus dans la réponse les object_list_elements de l'*object_list* qui ont un class_id égal à l'un des class_id de la class_list. Aucune information access_right n'est incluse. class_list::= array class_id class_id: long-unsigned

3	object_id_list	Accès par objet. L'enregistrement complet d'informations des instances d'objet sur l'object_id_list doit être retourné.
		object_id_list::= array object_id
		object_id::= structure
		{
		class_id: long-unsigned,
		logical_name: octet-string
		}

4	object_id	L'enregistrement complet d'informations de l'instance d'objet COSEM exigée doit être retourné. object_id::= structure Voir ci-dessus.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.3 associated_partners_id

Contient les identifiants des AP (de dispositif logique) client et serveur COSEM au sein des dispositifs physiques hébergeant ces AP, qui appartiennent à l'AA modélisée par l'objet "Association LN".

```
associated_partners_type::= structure
{
client_SAP:       integer,
server_SAP:       long-unsigned
}
```

La plage pour le client_SAP est 0…0x7F.

La plage pour le server_SAP est 0x0000…0x3FFF.

Les SAP doivent se situer dans la plage admise par le type de données et les supports.

4.4.4.2.4 application_context_name

Dans l'environnement COSEM, il est prévu qu'un contexte d'application préexiste et soit référencé par son nom pendant l'établissement d'une AA. Cet attribut contient le nom du contexte d'application pour l'AA en question.

```
context_name_type::=          CHOICE
{
context_name_structure              [2],
octet-string                        [9]
}
```

Le nom du contexte d'application est spécifié comme appartenant au type OBJECT IDENTIFIER dans l'IEC 62056-5-3:2023, 7.2.2.2.

Si le context_name_type est codé comme une structure, il contient les étiquettes d'arc ("arc labels") de l'OBJECT IDENTIFIER.

```
context_name_structure::= structure
{
joint_iso_ctt_element:                  unsigned,
country_element:                        unsigned,
country_name_element:                   long-unsigned,
identified_organization_element:        unsigned,
DLMS_UA_element:                        unsigned,
application_context_element:            unsigned,
context_id_element:                     unsigned
}
```

EXEMPLE 1: Dans le cas du context_id(1), le codage A-XDR est: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 01 11 01 (toutes les valeurs sont hexadécimales).

Si le context_name_type est codé comme une chaîne d'octets, il contient la valeur de l'OBJECT IDENTIFIER. Voir IEC 62056-5-3:2023, Article D.4.

EXEMPLE 2: Dans le cas du context_id(1), le codage A-XDR est: 09 07 60 85 74 05 08 01 01 (toutes les valeurs sont hexadécimales).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.5 xDLMS_context_info

Contient toutes les informations nécessaires sur le contexte xDLMS pour l'AA donnée.

```
xDLMS_context_type::= structure
{
conformance:                bit-string,
max_receive_pdu_size:       long-unsigned,
max_send_pdu_size:          long-unsigned,
dlms_version_number:        unsigned,
quality_of_service:         integer,
cyphering_info:             octet-string
}
```

Où:

– l'élément conformance (conformité) contient le bloc de conformité xDLMS pris en charge par le serveur. La longueur de la chaîne binaire est de 24 bits;

– L'élément max_receive_pdu_size contient la longueur maximale d'une APDU xDLMS, exprimée en octets, que le client peut envoyer, négociée pendant le processus d'association d'applications et limitée par le paramètre server-max-receive-pdu-size de l'APDU initiateResponse xDLMS;

– l'élément max_send_pdu_size, dans une AA active, contient la longueur maximale pour une APDU xDLMS, exprimée en octets, que le serveur peut envoyer. Elle est limitée par le paramètre client-max-receive-pdu-size de l'APDU initiateRequest xDLMS;

– l'élément dlms_version_number contient le numéro de version de DLMS® pris en charge par le serveur;

– l'élément quality_of_service n'est pas utilisé;

– l'élément cyphering_info, dans une AA active, contient le paramètre de clé dédié de l'APDU initiateRequest xDLMS. Voir IEC 62056-5-3:2023, Article 8.

4.4.4.2.6 authentication_mechanism_name

Contient le nom du mécanisme d'authentification pour l'AA.

```
    mechanism_name_type::=    CHOICE
{
    mechanism_name_structure        [2],
    octet-string                    [9]
}
```

Le nom du mécanisme d'authentification est spécifié sous la forme d'un OBJECT IDENTIFIER dans l'IEC 62056-5-3:2023, 7.2.2.3.

Si le mechanism_name_type est codé comme une structure, il contient les étiquettes d'arc ("arc labels") de l'OBJECT IDENTIFIER.

```
mechanism_name_structure::= structure
{
joint_iso_ctt_element:                  unsigned,
country_element:                        unsigned,
country_name_element:                   long-unsigned,
identified_organization_element:        unsigned,
DLMS_UA_element:                        unsigned,
authentication_mechanism_name_element:  unsigned,
mechanism_id_element:                   unsigned
}
```

EXEMPLE 3: Dans le cas du mechanism_id(1), le codage A-XDR est: 02 07 11 02 11 10 12 02 F4 11 05 11 08 11 02 11 01 (toutes les valeurs sont hexadécimales).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si le mechanism_name_type est codé comme une chaîne d'octets, il contient la valeur de l'OBJECT IDENTIFIER. Voir IEC 62056-5-3:2023, Article D.4.

EXEMPLE 4: Dans le cas du mechanism_id(1), le codage A-XDR est: 09 07 60 85 74 05 08 02 01 (toutes les valeurs sont hexadécimales).

Aucun mechanism_name n'est exigé lorsqu'aucune authentification n'est utilisée.

4.4.4.2.7 secret

Contient le secret pour le processus d'authentification LLS ou HLS.

NOTE Dans le cas de HLS avec GMAC, le secret (HLS) est détenu par l'objet "Security setup" référencé dans l'attribut 9, *security_setup_ reference, 4.4.4.2.9*.

4.4.4.2.8 association_status

Indique l'état courant de l'association qui est modélisée par l'objet.

enum:

(0) non associé, association en attente,
(1) associé

4.4.4.2.9 security_setup_ reference

Référence un objet "Security setup" par son nom logique. L'objet référencé gère la sécurité pour une instance donnée de l'objet "Association LN".

4.4.4.2.10 user_list

Contient la liste des utilisateurs autorisés à utiliser l'AA gérée par l'instance donnée de l'IC "Association LN".

```
array        user_list_entry

user_list_entry::=      structure
{
user_id:                unsigned,

  user_name             CHOICE
                        {
                        visible-string      [10],
                        UTF-8-string        [12]
                        }
}
```

Où:

user_id est l'identifiant de l'utilisateur (cette valeur se trouve dans le champ calling-AE-invocation-id de l'AARQ);

user_name est le nom de l'utilisateur.

Lorsque l'attribut *user_list* est vide (c'est-à-dire un tableau de 0 élément), tous les utilisateurs peuvent utiliser l'AA, c'est-à-dire que le champ calling-AE-invocation-id de l'AARQ est ignoré.

Lorsque l'attribut *user_list* n'est pas vide, seuls les utilisateurs présents dans la liste peuvent établir l'AA, c'est-à-dire que le champ calling-AE-invocation-id de l'AARQ doit être présent et que sa valeur doit correspondre à l'un des user_ids de l'*user_list*, sinon l'AA n'est pas établie.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.2.11 current_user

Contient l'identifiant de l'utilisateur en cours.

current_user::= user_list_entry (voir 4.4.4.2.10)

Si *user_list* est vide, current_user doit être une structure {user_id: unsigned 0, user_name: chaîne visible de 0 élément}.

4.4.4.3 Description de la méthode

4.4.4.3.1 reply_to_HLS_authentication (data)

L'appel à distance de cette méthode délivre au serveur le résultat du traitement de secret par le client du défi lancé par le serveur au client, f(StoC), comme le paramètre de service *data* (données) de la primitive ACTION.request appelée.

data::= octet-string réponse du client au défi

Si l'authentification est acceptée, la réponse (primitive ACTION.confirm) contient Result == OK et le résultat du traitement de secret par le serveur du défi du client au serveur, f(CtoS) dans le paramètre de service *data* du service de réponse.

data::= octet-string réponse du serveur au défi

Si l'authentification n'est pas acceptée, le paramètre *result* dans la réponse doit contenir une valeur non-OK et aucune donnée ne doit être renvoyée.

4.4.4.3.2 change_HLS_secret (data)

Change le secret HLS (par exemple: clé de cryptage).

data::= octet-string nouveau secret HLS

La structure du "new secret" dépend du mécanisme de sécurité mis en œuvre. Le "new secret" peut contenir des bits de contrôle supplémentaires et il peut être crypté.

4.4.4.3.3 add_object (data)

Ajoute l'objet référencé à l'*object_list*.

data::= object_list_element (voir 4.4.4.2.2)

4.4.4.3.4 remove_object (data)

Retire l'objet référencé de l'*object_list*.

data::= object_list_element (voir 4.4.4.2.2)

4.4.4.3.5 add_user (data)

Ajoute un utilisateur à l'*user_list*.

data::= user_list_entry (voir 4.4.4.2.10)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.4.3.6 remove_user (data)

Supprime un utilisateur de l'*user_list*.

data::= user_list_entry (voir 4.4.4.2.10)

4.4.5 SAP assignment (class_id = 17, version = 0)

4.4.5.1 Vue d'ensemble

Cette IC permet de modéliser la structure logique des dispositifs physiques, en fournissant des informations relatives à l'affectation des dispositifs logiques à leurs SAP. Voir IEC 62056-5-3:2023, Annexe A.

SAP assignment		**0...1**	**class_id = 17, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. SAP_assignment_list	(static)	asslist_type			0	x + 0x08
Méthodes spécifiques		***o/f***				
1. connect_logical_device (data)		f				x + 0x20

4.4.5.2 Description d'attribut

4.4.5.2.1 logical_name

Identifie l'instance de l'objet "SAP assignment". Voir 6.2.34.

4.4.5.2.2 SAP_assignment_ list

Contient la liste de tous les dispositifs logiques et leurs adresses de SAP au sein du dispositif physique.

```
asslist_type::= array          asslist_element
asslist_element::= structure
{
SAP:                    long-unsigned,
logical_device_name:    CHOICE
                {
                octet-string        [9],
                visible-string      [10],
                utf8-string         [12]
                }
}
```

NOTE L'adressage effectif est réalisé par les couches de communication de support.

4.4.5.3 Méthode connect_logical_ device (data)

Connecte un dispositif logique à un SAP. La connexion au SAP 0 déconnectera le dispositif. Plusieurs dispositifs ne peuvent pas être connectés à un SAP (à l'exception du SAP 0).

data::= asslist_element

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6 Image transfer (class_id = 18, version = 0)

4.4.6.1 Généralités

Les instances de l'IC "Image transfer" modélisent le processus de transfert de fichiers binaires, appelés Images, vers les serveurs COSEM.

NOTE La présente spécification inclut certaines améliorations et précisions apportées au texte. Les principales modifications sont les suivantes:

- le processus de transfert d'images est décrit par le biais d'un seul exemple. Le texte et l'organigramme ont été mis à jour;
- l'échange de données entre le client et un serveur d'image conceptuel est hors du domaine d'application;
- l'image transférée et l'image à activer sont clairement distinguées;
- les étapes 1 et 6 sont à présent facultatives;
- la taille de la chaîne binaire *image_transferred_blocks_status* peut être dynamique;
- désormais, il est indiqué que le fait d'attribuer la valeur FALSE à l'attribut *image_transfer_enabled* désactive le processus de transfert d'images;
- il est également spécifié que la réinitialisation du processus de transfert d'images réinitialise désormais l'ensemble du processus;
- certaines précisions ont été ajoutées aux effets de l'appel des méthodes.

4.4.6.2 Étapes du processus de transfert d'images

Le transfert d'images se déroule généralement en plusieurs étapes:

- étape 1: (facultative): récupérer l'ImageBlockSize;
- étape 2: le client lance le transfert d'images;
- étape 3: le client transfère les ImageBlocks;
- étape 4: le client vérifie l'exhaustivité de l'Image;
- étape 5: le serveur vérifie l'Image (initiée par le client ou de son propre chef);
- étape 6 (facultative): le client vérifie les informations relatives aux images à activer;
- étape 7: le serveur active la ou les images (initiée par le client ou de son propre chef).

Pour obtenir un exemple avec des explications plus détaillées, voir 4.4.6.6.

4.4.6.3 Vue d'ensemble

Cet IC permet de fournir les attributs et les méthodes avec lesquels modéliser le processus de transfert d'image.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Image transfer		**0...n**	**class_id = 18, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. image_block_size	(static)	double-long-unsigned				x + 0x08
3. image_transferred_blocks_status	(dyn.)	bit-string				x + 0x10
4. image_first_not_transferred_block_number	(dyn.)	double-long-unsigned				x + 0x18
5. image_transfer_enabled	(static)	boolean				x + 0x20
6. image_transfer_status	(dyn.)	enum				x + 0x28
7. image_to_activate_info	(dyn.)	array				x + 0x30
Méthodes spécifiques		***o/f***				
1. image_transfer_initiate (data)		o				x + 0x40
2. image_block_transfer (data)		o				x + 0x48
3. image_verify (data)		o				x + 0x50
4. image_activate (data)		o				x + 0x58

4.4.6.4 Description d'attribut

4.4.6.4.1 logical_name

Identifie l'instance de l'objet "Image transfer". Voir 6.2.37.

4.4.6.4.2 image_block_size

Contient l'ImageBlockSize, en octets, qui peut être gérée par le serveur. ImageBlockSize ne doit pas dépasser la ServerMaxReceivePduSize négociée.

NOTE *image_block_size* est une propriété du serveur.

4.4.6.4.3 image_transferred_blocks_status

Fournit les informations relatives à l'état du transfert de chaque ImageBlock. Chaque bit dans la chaîne binaire fournit des informations relatives à un ImageBlock pris séparément:

0 = non transféré;

1 = transféré

NOTE La taille de l'attribut peut être dynamique, c'est-à-dire qu'elle peut croître à réception de nouveaux ImageBlocks.

4.4.6.4.4 image_first_not_transferred_block_number

Fournit l'ImageBlockNumber du premier ImageBlock non transféré. Une fois que l'Image est complète, il convient que la valeur retournée soit supérieure ou égale au nombre de blocs calculé à partir de la taille de l'Image et de l'ImageBlockSize.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.4.5 image_transfer_enabled

Contrôle l'activation du processus de transfert d'images.

boolean:

FALSE = désactivé,
TRUE = activé

Les méthodes de transfert d'images peuvent être appelées uniquement si la valeur de cet attribut est TRUE. Si la valeur FALSE lui est attribuée, toutes les méthodes sont désactivées (échec de l'appel).

4.4.6.4.6 image_transfer_status

Contient l'état du processus de transfert d'images.

enum:

(0) transfert d'images non déclenché;
(1) transfert d'images déclenché;
(2) vérification d'image lancée;
(3) vérification d'image réussie;
(4) vérification d'image échouée;
(5) activation d'image lancée;
(6) activation d'image réussie;
(7) activation d'image échouée.

4.4.6.4.7 image_to_activate_info

Fournit les informations de l'Image ou des Images prêtes à l'activation. Il est généré comme le résultat du processus de vérification de l'Image. Le client peut vérifier ces informations avant d'activer les Images.

```
array       image_to_activate_info_element

image_to_activate_info_element::= structure
{
image_to_activate_size:              double-long-unsigned,
image_to_activate_identification:    octet-string,
image_to_activate_signature:         octet-string
}
```

Où:

image_to_activate_size	est la taille de l'Image à activer, exprimée en octets;
image_to_activate_identification	est l'identification de l'Image à activer et peut contenir des informations telles que le fabricant, le type de dispositif, les informations de version, etc.;
image_to_activate_signature	est la signature de l'Image à activer.

NOTE L'algorithme de génération de la signature est hors du domaine d'application de la présente spécification.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.5 Description de la méthode

4.4.6.5.1 image_transfer_initiate (data)

Initialise le processus de transfert d'images.

```
data::= structure
{
    image_identifier:    octet-string,
    image_size:          double-long-unsigned
}
```

Où:

– image_identifier identifie l'Image à transférer;

– image_size contient l'ImageSize, en octets.

NOTE L'*image_identifier* identifie l'Image à transférer (conteneur), mais n'est pas nécessairement lié à son contenu, c'est-à-dire les Images qui seront activées. Ces informations peuvent être récupérées de l'attribut *image_to_activate* après vérification de l'Image transférée.

À la suite de l'appel réussi de la méthode, la valeur (1) est affectée à l'attribut *image_transfer_status* et la valeur 0 à *image_first_not_transferred_block_number*. Tous les appels suivants de la méthode réinitialisent l'ensemble du processus de transfert d'images, et tous les ImageBlocks sont de nouveau à transférer.

4.4.6.5.2 image_block_transfer (data)

Transfère un bloc de l'Image vers le serveur.

```
data::= structure
{
image_block_number:    double-long-unsigned,
image_block_value:     octet-string
}s
```

À la suite de l'appel réussi de la méthode, la valeur 1 est affectée au bit correspondant de l'attribut *image_transferred_blocks_status* et l'attribut *image_first_not_transferred_block_number* est mis à jour.

4.4.6.5.3 image_verify (data)

Vérifie l'intégrité de l'Image avant activation.

```
data::= integer (0)
```

Le résultat de l'appel de cette méthode peut être "success", "temporary_failure" ou "other_reason". S'il n'est pas "success", le résultat de la vérification peut être déterminé en récupérant la valeur de l'attribut *image_transfer_status*.

En cas de succès, l'attribut *image_to_activate_info* contient les informations relatives aux images à activer.

NOTE Pour les services xDLMS, les codes Action-Result/Data-Access-Result sont spécifiés dans l'IEC 62056-5-3:2023, Article 8.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.5.4 image_activate (data)

Active l'Image.

data::= integer (0)

Si l'Image transférée n'a pas été vérifiée au préalable, cette vérification est effectuée dans le cadre de l'activation de l'Image.

Le résultat de l'appel de cette méthode peut être "success", "temporary-failure" ou "other-reason". S'il n'est pas "success", le résultat de l'activation peut être connu en récupérant la valeur de l'attribut *image_transfer_status*.

NOTE Pour les services xDLMS, les codes Action-Result/Data-Access-Result sont spécifiés dans l'IEC 62056-5-3:2023, Article 8.

4.4.6.6 Exemple de processus de transfert d'images normalisé

4.4.6.6.1 Généralités

Le présent paragraphe et l'organigramme de la Figure 11 décrivent un exemple de processus courant de transfert d'images à partir d'un client présumé. À noter qu'il n'est pas obligatoire de suivre ce processus. Selon le cas d'utilisation, certaines étapes peuvent être ignorées alors que d'autres peuvent être nécessaires.

4.4.6.6.2 Condition préalable:

Le transfert d'Image doit être activé: *image_transfer_enabled* = TRUE.

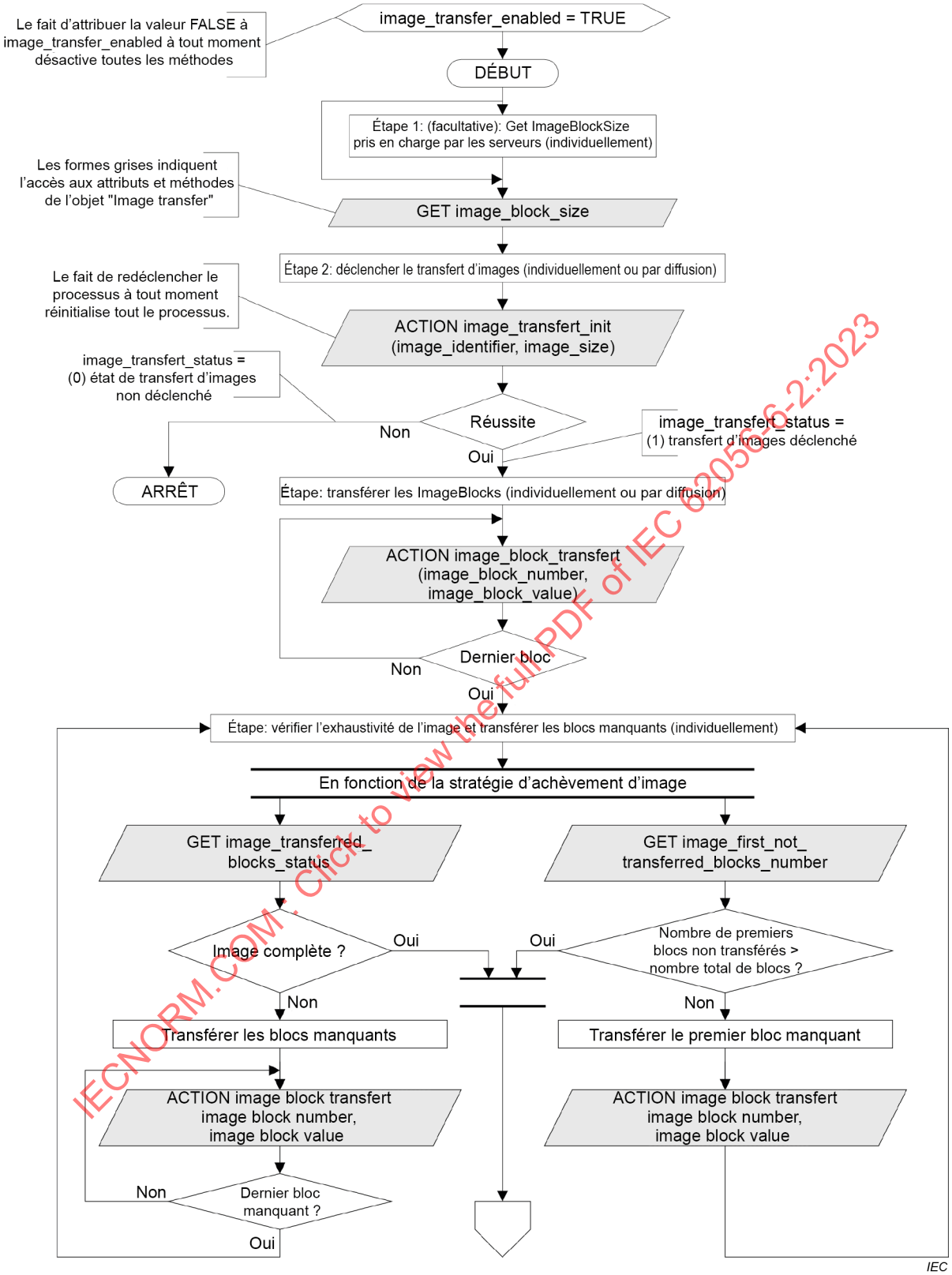
Le fait d'attribuer la valeur FALSE à *image_transfer_enabled* à tout moment désactive toutes les méthodes (l'appel de ces méthodes échoue). La valeur des attributs d'état n'est pas définie.

4.4.6.6.3 Étape 1 (facultative): récupérer l'ImageBlockSize

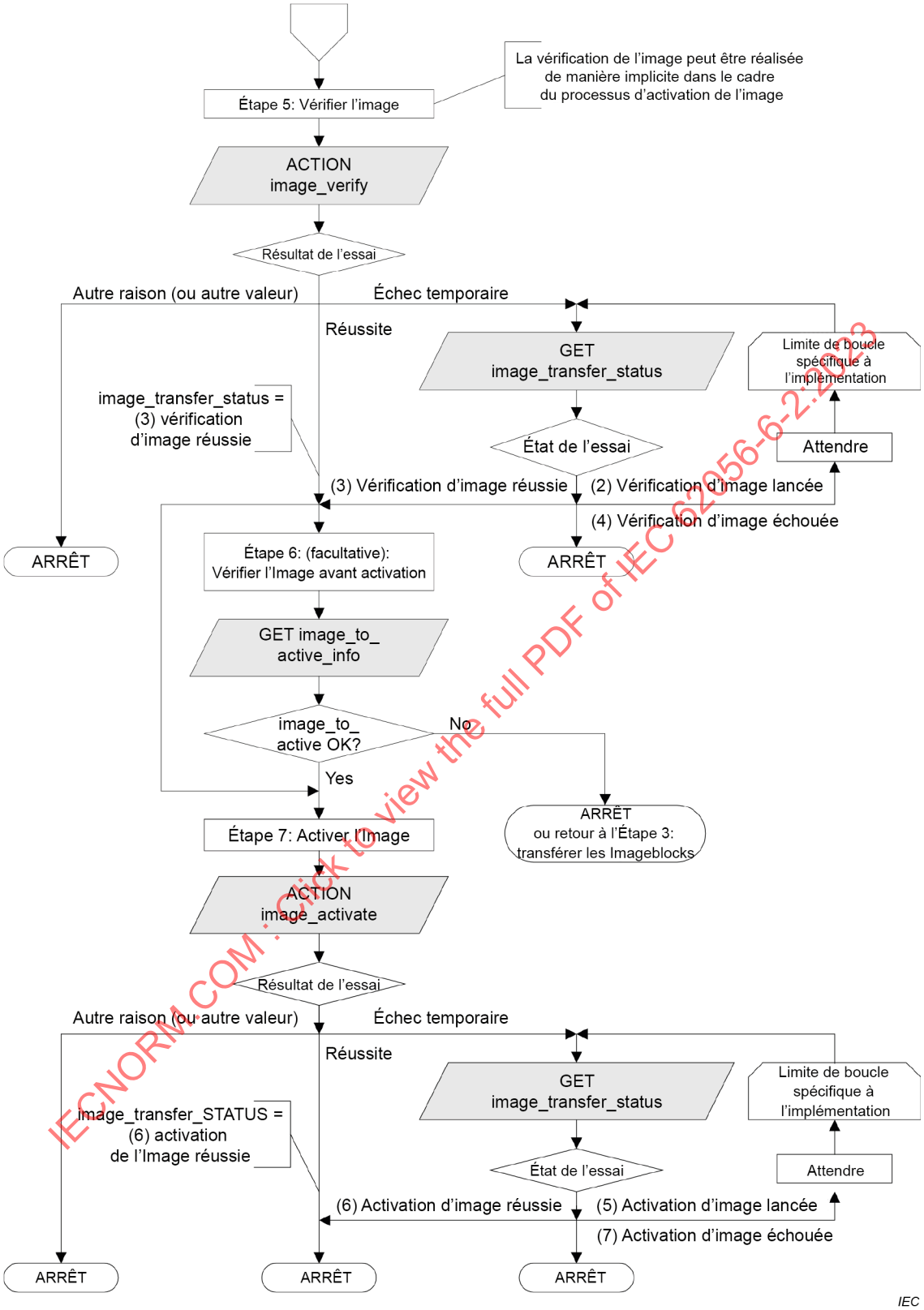
Si le client ne connaît pas la taille des blocs d'image, le serveur cible du transfert d'images peut la traiter. Il doit lire l'attribut *image_block_size* de l'objet "Image transfer" correspondant de chaque serveur vers lequel l'image doit être transférée, et ce avant de lancer le processus. Le client peut alors transférer les ImageBlocks de la bonne taille.

Si les ImageBlocks sont envoyés, par diffusion, vers un groupe de serveurs COSEM, l'ImageBlockSize doit être le même dans chaque membre du groupe.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 11 – Organigramme du processus de transfert d'images

4.4.6.6.4 Étape 2: le client lance le transfert d'Image

Le client lance le processus de transfert d'images individuellement ou par diffusion dans tous les serveurs, en appelant la méthode *image_transfer_initiate*. Le paramètre d'appel de la méthode contient l'identifiant et la taille de l'Image à transférer. Le serveur doit rendre disponible l'espace mémoire nécessaire pour contenir l'Image.

Après un lancement réussi, la valeur de l'attribut *image_transfer_status* est (1). L'attribut *image_transferred_blocks_status* doit être réinitialisé, la valeur 0 doit être affectée à l'attribut *image_first_not_transferred_block_number* et il convient de réinitialiser la valeur de l'attribut *image_to_activate_info*. Le processus de transfert d'images est lancé, et le serveur COSEM se prépare à accepter les ImageBlocks.

4.4.6.6.5 Étape 3: le client transfère les ImageBlocks

Le client transfère les ImageBlocks vers un (groupe de) serveur(s) en appelant la méthode *image_block_transfer* individuellement ou par diffusion. Les paramètres d'appel de la méthode comprennent l'ImageBlockNumber et un ImageBlock. Les ImageBlocks sont acceptés par les seuls serveurs COSEM dans lesquels le processus de transfert d'Image a été déclenché avec succès. Les autres serveurs rejettent silencieusement tous les éventuels ImageBlocks reçus.

4.4.6.6.6 Étape 4: le client vérifie l'exhaustivité de l'Image

Le client vérifie, pour chaque serveur individuellement, l'exhaustivité de l'Image transférée. Si l'Image n'est pas complète, il transfère les ImageBlocks non (encore) transférés. Ce processus itératif se poursuit jusqu'au transfert réussi de l'Image complète.

Pour identifier et transférer les ImageBlocks non transférés, deux mécanismes sont disponibles.

- Le client peut récupérer l'état de chaque ImageBlock: "non transféré" ou "transféré". Ceci est réalisé en récupérant la valeur de l'attribut *image_transferred_blocks_status*. Le client transfère ensuite les ImageBlocks non (encore) transférés.
- En variante, le client peut récupérer l'ImageBlockNumber du premier bloc non transféré. Dans ce cas, il récupère la valeur de l'attribut *image_first_not_transferred_block_number*. Le client transfère alors l'ImageBlock qui n'est pas (encore) transféré.
- Le client vérifie ensuite de nouveau l'exhaustivité de l'Image.

NOTE Les deux mécanismes peuvent être combinés librement.

4.4.6.6.7 Étape 5: le serveur vérifie l'Image

L'Image est vérifiée par le serveur. À cet effet, le client appelle la méthode *image_verify* ou le serveur peut lancer la vérification. Le résultat peut être:

- un succès ("success"), si la vérification a pu être achevée;
- un échec temporaire ("temporary-failure"), si la vérification n'a pas pu être achevée;
- une autre raison ("other-reason"), si la vérification a échoué.

NOTE Les conditions de vérification de l'Image ne relèvent pas du domaine d'application du présent document.

4.4.6.6.8 Étape 6 (facultative): le client vérifie les informations relatives aux images à activer

Le résultat de la vérification de l'Image peut être vérifié par le client en récupérant la valeur de l'attribut *image_transfer_status*. La valeur de cet attribut est mise à jour à la suite de la vérification de l'Image.

Une Image transférée peut contenir une ou plusieurs Images à activer. Pour chaque Image à activer, l'attribut image_to_activate_info contient les paramètres: {image_to_activate_size, image_to_activate_identification, image_to_activate_signature}.

Si ces informations ne sont pas celles attendues, le client peut recommencer le transfert de l'image.

Sinon il passe à l'étape suivante d'activation de l'Image.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.6.6.9 Étape 7: le serveur active la ou les Images

L'Image est activée par le serveur. À cet effet, le client appelle la méthode *image_activate* ou le serveur peut lancer l'activation. Si l'activation est effectuée sans vérification préalable, une vérification est implicitement réalisée dans le cadre du processus d'activation. Le résultat de l'appel de la méthode *Image_activate* peut être:

- un succès ("success"), si l'activation d'Image a abouti;
- un échec temporaire ("temporary-failure"), si la vérification/activation n'a pas pu être achevée;
- une autre raison ("other-reason"), si l'activation a échoué.

En cas de succès, le serveur procède à l'activation de la ou des nouvelles Images. Au cours de ce processus, il n'est pas accessible. Après l'activation de la ou des Images, le résultat peut être vérifié par le client, en récupérant la valeur de l'attribut *image_transfer_status* ou en lisant le contenu des objets COSEM appropriés contenant l'identifiant, la version et la signature numérique du progiciel actif.

4.4.7 Security setup (class_id = 64, version = 1)

4.4.7.1 Vue d'ensemble

Les instances de l'IC "Security setup" contiennent les informations nécessaires à la suite de sécurité utilisée et concernant la politique de sécurité applicable entre un client et un serveur, identifiés par leur titre de système respectif. Elles fournissent également les méthodes permettant d'élever le niveau de sécurité et de gérer les clés symétriques, les paires de clés asymétriques et les certificats.

Security setup		**0…n**	**class_id = 64, version = 1**			
Attributs		**Type de données**	**Min.**	**Max.**	**Def.**	**Nom court**
1. logical_name	(static)	octet-string				x
2. security_policy	(static)	enum				x + 0x08
3. security_suite	(static)	enum				x + 0x10
4. client_system_title	(dyn.)	octet-string				x + 0x18
5. server_system_title	(static)	octet-string				x + 0x20
6. certificates	(dyn.)	array				x + 0x28
Méthodes spécifiques		o/f				
1. security_activate (data)		f				x + 0x30
2. key_transfer (data)		f				x + 0x38
3. key_agreement (data)		f				x + 0x48
4. generate_key_pair (data)		f				x + 0x50
5. generate_certificate_request (data)		f				x + 0x58
6. import_certificate (data)		f				x + 0x60
7. export_certificate (data)		f				x + 0x68
8. remove_certificate (data)		f				x + 0x70

4.4.7.2 Description d'attribut

4.4.7.2.1 logical_name

Identifie l'instance de l'objet "Security setup". Voir 6.2.36.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.7.2.2 security_policy

Procède à l'authentification et/ou au cryptage et/ou à la signature numérique en utilisant les algorithmes de sécurité disponibles dans la suite de sécurité.

Elle s'applique indépendamment des demandes et des réponses.

enum: Si la valeur enum est interprétée comme un unsigned, la signification de chaque bit est la suivante:

Bit	Politique de sécurité
0	non utilisé, doit être positionné à 0;
1	non utilisé, doit être positionné à 0;
2	demande authentifiée;
3	demande cryptée;
4	demande signée numériquement;
5	réponse authentifiée;
6	réponse cryptée;
7	réponse signée numériquement.

NOTE La valeur (0) signifie qu'aucune protection cryptographique n'est exigée, comme dans la version 0 de l'IC "Security setup".

Les droits d'accès peuvent exiger une protection plus importante que ne l'exige la politique de sécurité.

4.4.7.2.3 security_suite

Spécifie les algorithmes de sécurité disponibles.

enum:

(0) cryptage authentifié AES-GCM-128 et enveloppement de clés AES-128;
(1) cryptage authentifié AES-GCM-128, signature numérique ECDSA P-256, agrément de clé ECDH P-256, hachage SHA-256, compression V.44 et enveloppement de clés AES-128;
(2) cryptage authentifié AES-GCM-256, signature numérique ECDSA P-384, agrément de clé ECDH P-384, hachage SHA-384, compression V.44 et enveloppement de clés AES-256;
(3) … (15) réserves.

4.4.7.2.4 client_system_title

Transporte le titre de système du client (courant):

– dans l'environnement PLC S-FSK, l'initiateur actif envoie son titre de système en utilisant le protocole CIASE;

NOTE Il est également détenu par l'attribut active_initiator de l'objet "S-FSK Active initiator" (voir 4.10.4).

– au cours de l'établissement d'AA confirmé ou non confirmé, il est transporté par le champ calling-AP-title de l'APDU d'AARQ.

  Si un titre de système du client a déjà été envoyé pendant un processus d'enregistrement, comme dans le cas du profil PLC S-FSK, il convient que le titre de système du client transporté par l'APDU d'AARQ soit le même. Sinon l'AA doit être rejetée et il convient d'envoyer des informations de diagnostic appropriées;

– dans une AA préétablie, il peut être écrit par le client avec l'aide d'un service non sécurisé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.7.2.5 server_system_title

Transporte le titre de système du serveur.

– Dans l'environnement PLC S-FSK, le serveur envoie son titre de système pendant le processus de découverte, en utilisant le protocole CIASE;
– au cours de l'établissement d'AA confirmé, il est transporté par le champ responding-AP-title de l'APDU d'AARE.

Cet attribut doit être en lecture seulement.

4.4.7.2.6 certificates

Transporte les certificats X.509 v3 disponibles et stockés sur le serveur.

```
array                certificate_info

certificate_info::= structure
{
certificate_entity: enum:
                                    (0) serveur;
                                    (1) client;
                                    (2) autorité de certification;
                                    (3) autre.
certificate_type: enum:
                                    (0) signature numérique;
                                    (1) agrément de clé;
                                    (2) TLS;
                                    (3) autre.
        serial_number:      octet-string;
        issuer:             octet-string;
        subject:            octet-string;
        subject_alt_name:   octet-string.
}
```

Pour les éléments serial_number, issuer, subject, subject_alt_name voir IEC 62056-5-3:2023, 5.6.4.

Un serveur qui utilise le cryptage par clé publique stocke les certificats de clé publique suivants:

pour le serveur:

– certificat de signature numérique;
– certificat d'agrément de clé statique;
– certificat TLS;

pour chaque client et tiers:

– certificat de signature numérique;
– certificat d'agrément de clé statique;
– certificat TLS;

pour les autorités de certification:

– certificat d'autorité de certification.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.7.3 Description de la méthode

4.4.7.3.1 security_activate (data)

Active et renforce la politique de sécurité:

Data::= enum, tel que spécifié dans l'attribut *security_policy*.

Le renforcement de la politique de sécurité signifie qu'un autre type de protection est ajouté. Une fois ajouté, ce type de protection ne peut plus être retiré.

Une fois ajouté, ce type de protection ne peut plus être retiré. Si la méthode est appelée avec une valeur indiquant une politique de sécurité plus faible que la valeur de l'attribut security_policy, l'appel de la méthode n'aboutit pas.

La nouvelle politique de sécurité s'applique dès que l'appel de la méthode a été confirmé avec succès.

4.4.7.3.2 key_transfer (data)

Permet de transférer une ou plusieurs clés symétriques.

Le paramètre *data* inclut l'identification de la ou des clés transportées et de la ou des clés enveloppées elles-mêmes.

```
data::=        array key_transfer_data

key_transfer_data::= structure
{
        key_id:        enum:
                              (0) clé globale de cryptage monodiffusion,
                              (1) clé globale de cryptage de diffusion,
                              (2) clé d'authentification,
                              (3) clé principale (KEK)

        key_wrapped:   octet-string
}
```

NOTE 1 Il est possible de transférer plusieurs clés en appelant la méthode avec un tableau contenant *nkey_transfer data*, ou d'appeler la méthode *n* fois avec un tableau contenant un seul *key_transfer_data*.

L'algorithme d'enveloppement de clé est tel que spécifié par la suite de sécurité. Dans les suites 0, 1 et 2, l'algorithme d'enveloppement de clé AES est utilisé.

La KEK est la clé principale.

Si le développement de la ou des clés aboutit, le résultat de l'appel de la méthode aboutit également.

Si le développement de la ou des clés n'aboutit pas, l'appel de la méthode échoue et cet échec doit être justifié. Les clés ne sont pas modifiées.

Les nouvelles clés sont activées immédiatement après le renvoi du résultat de l'appel de la méthode avec "result = success". Noter que cette règle s'applique à toutes les clés, y compris la clé principale.

NOTE 2 Les APDU contenant les primitives de service d'appel de la méthode sont protégées comme indiqué par *security_policy* et par les droits d'accès aux méthodes. En outre, les paramètres d'appel de la méthode peuvent être protégés en appelant la méthode par l'intermédiaire d'un objet "Data protection". La protection exigée peut être stipulée dans les spécifications auxiliaires spécifiques à un projet donné.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Cette note s'applique également à la méthode *key_agreement* (data).

NOTE 3 En cas d'échec de l'utilisation des nouvelles clés, le client peut tenter de remédier à cette situation en revenant aux clés précédentes ou en répétant le transfert de clé.

4.4.7.3.3 key_agreement (data)

Permet de convenir d'une ou plusieurs clés symétriques en utilisant l'algorithme d'agrément de clé spécifié par la suite de sécurité. Dans le cas des suites 1 et 2, l'algorithme d'agrément de clé ECDH est utilisé avec le schéma Ephemeral Unified Model C(2e, 0s, ECC CDH).

Le paramètre *data* inclut l'identification de la ou des clés et des données utilisées dans la transaction d'agrément de clé.

```
data::=      array key_agreement_data

key_agreement_data::= structure
{
      key_id       enum:
            (0) clé globale de cryptage monodiffusion,
            (1) clé globale de cryptage de diffusion,
            (2) clé d'authentification,
            (3) clé principale (KEK)
      key_data:   octet-string
}
```

NOTE 1 Il est possible de convenir de plusieurs clés en appelant la méthode avec un tableau contenant *n key_agreement_data*, ou d'appeler la méthode *n* fois avec un tableau contenant un seul *key_agreement_data*.

L'élément key_id identifie la ou les clés à convenir.

L'élément *key_data* est la clé publique de la clé éphémère concaténée à droite avec une signature numérique, qui est calculée sur la valeur key_id et la clé publique de la valeur de paire de clés publiques éphémères à l'aide de la clé privée de signature numérique: $Q_{e,U}$, II ECDSA(key_id II $Q_{e,U}$).

Si le serveur peut vérifier la signature numérique de *key_data*, calculer le secret partagé et déduire la clé, alors l'appel de la méthode aboutit et le serveur envoie sa *key_data* au client. Sinon, l'appel de la méthode échoue et la raison de l'échec est renvoyée.

Les nouvelles clés sont activées immédiatement après le renvoi du résultat de l'appel de la méthode avec "result = success".

NOTE 2 En cas d'échec de l'utilisation des nouvelles clés, le client peut tenter de remédier à cette situation en revenant aux clés précédentes ou en répétant le transfert de clé.

4.4.7.3.4 generate_key_pair (data)

Génère une paire de clés asymétriques comme exigé par la suite de sécurité. Le paramètre *data* identifie l'usage de la paire de clés à générer.

```
data  ::= enum:

(0)  paire de clés à signature numérique;
(1)  paire de clés à agrément de clé;
(2)  paire de clés TLS.
```

NOTE 1 Il y a au maximum une paire de clés par signature numérique, par agrément de clé et par TLS.

NOTE 2 La paire de clés à agrément de clé est la clé statique utilisée avec le schéma One-Pass Diffie-Hellman C(1e, 1s, ECC CDH) lorsque le serveur joue le rôle de la Partie V ou avec le schéma Static Unified Model C(0e, 2s, ECC CDH).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.7.3.5 generate_certificate_request (data)

Si cette méthode est appelée, le serveur envoie les données de demande de signature de certificat (CRS) dont a besoin l'autorité de certification pour générer un certificat correspondant à une clé publique de serveur. Le paramètre *data* identifie la paire de clés pour laquelle le certificat sera demandé.

data ::= enum:

(0) paire de clés à signature numérique;
(1) paire de clés à agrément de clé;
(2) paire de clés TLS.

NOTE 1 Il y a au maximum une paire de clés par signature numérique, par agrément de clé et par TLS.

Les paramètres de réponse d'appel de la méthode incluent les données nécessaires à la génération du certificat.

NOTE 2 Leur transfert à l'autorité de certification relève de la responsabilité du client.

response data::= octet-string,

La chaîne d'octets contient le codage DER de la DSR comme indiqué dans PKCS #10 (RFC 2986).

La CSR doit être signée par la clé privée appartenant à la clé publique dans la demande. Elle fait office de "preuve de possession" de la clé privée.

4.4.7.3.6 import_certificate (data)

Importe un certificat X.509 v3 d'une clé publique.

data ::= octet-string

La chaîne d'octets contient le codage DER de la DSR comme indiqué dans PKCS #10 (RFC 2986).

4.4.7.3.7 export_certificate (data)

Exporte un certificat X.509 v3 dans le serveur.

Le certificat est identifié par une identification d'entité ou par son numéro de série.

```
data::= certificate_identification

certificate_identification::= structure
{
        certificate_identification_type:        enum:
                                        (0) certificate_identification_entity,
                                        (1) certificate_identification_serial

        certification_identification_options: CHOICE
        {
                certificate_identification_by_entity,
                certificate_identification_by_serial
}
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
certificate_identification_by_entity::= structure
{
        certificate_entity: enum:
                                (0) serveur,
                                (1) client,
                                (2) autorité de certification,
                                (3) autre
        certificate_type: enum:
                                (0) signature numérique,
                                (1) agrément de clé,
                                (2) TLS
                                (3) autre

        system_title:           octet-string
}

certificate_identification_by_serial::= structure
{
        serial_number:          octet-string,
        issuer:                 octet-string
}

response data::= octet-string
La chaîne d'octets de données de réponse est mise au format DER X.509 v3.
```

En l'absence de certificat correspondant, la réponse doit contenir un code d'erreur approprié.

4.4.7.3.8 remove_certificate (data)

Retire un certificat X.509 v3 du serveur.

```
data::= certificate_identification

certificate_identification (voir 4.4.7.3.7).
```

NOTE Les certificats du serveur pour la signature numérique, l'agrément de clé et TLS ne peuvent pas être retirés du serveur. Lorsqu'une mise à jour de ces certificats est nécessaire, la nouvelle paire de clés est générée, une nouvelle demande de certificat est générée et un nouveau certificat est importé.

4.4.8 Classes d'interfaces et objets Push

4.4.8.1 Vue d'ensemble

Dans plusieurs cas, les messages DLMS® peuvent être transmis par opération Push vers une destination sans être demandés de manière explicite. L'envoi Push est initié par une condition de déclenchement comme suit:

- si un temps de référence a été atteint;
- si une valeur surveillée en local a dépassé un seuil;
- un événement local (mise sous/hors tension, pression sur le bouton-poussoir, couvercle du compteur ouvert, par exemple).

Le mécanisme d'envoi Push DLMS®/COSEM suit le modèle de publication/abonnement:

La publication est modélisée dans DLMS®/COSEM par l'attribut *object_list* des objets "Association" qui fournit la liste des objets COSEM et de leurs attributs accessibles dans une AA donnée. L'abonnement est modélisé par l'écriture des attributs appropriés des objets "Push setup". Les données exigées sont envoyées (lors du déclenchement spécifié) à l'aide du service xDLMS DataNotification lorsque le déclenchement spécifié est activé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

La Figure 12 illustre le modèle COSEM de l'opération Push.

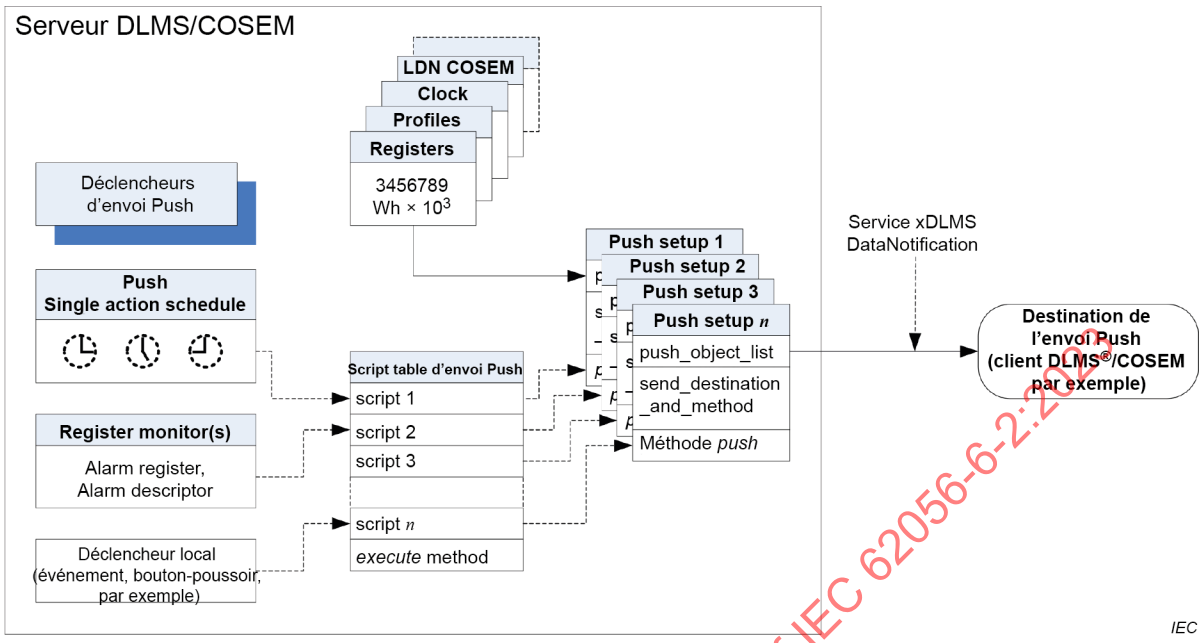


Figure 12 – Modèle COSEM d'opération Push

L'élément central de la modélisation de l'opération Push est l'IC "Push setup". L'attribut *push_object_list* contient une liste des références aux attributs d'objet COSEM à transmettre par opération Push.

Lorsque le mécanisme d'envoi Push utilise une passerelle, alors la version du protocole de passerelle pour les dispositifs finaux sans connaissance du WAN/NN doit être appliquée. Ceci est décrit dans l'IEC 62056-5-3:2023, 10.9.4.3.

Les différents déclencheurs (programmateurs, moniteurs, déclencheurs locaux, etc.) appellent une entrée de script dans un objet d'envoi Push "Script table" (nouvelles instances de l'IC "Script table") qui appelle alors la méthode *push* de l'objet "Push setup" correspondant. La destination, le support de communication, le protocole, le codage, la synchronisation et les nouvelles tentatives d'opération Push sont déterminés par les autres attributs de l'objet "Push setup".

Chaque déclencheur peut être à l'origine de l'envoi des données à une destination dédiée. Par conséquent, pour chaque déclencheur ou groupe de déclencheurs, un objet "Push setup" individuel est disponible pour définir le contenu et la destination du message d'envoi Push, ainsi que le support de communication utilisé.

Pour les besoins de l'envoi Push des données lorsqu'une alarme se déclenche, les objets Alarm monitor (nouvelles instances de l'IC "Register monitor") sont disponibles.

Les alarmes sont détenues par les objets *Alarm register* ou *Alarm descriptor* (voir 6.2.64).

La structure de l'objet *Alarm descriptor* et les conditions d'alarme nécessitent d'être définies dans une spécification d'accompagnement spécifique au projet.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Les objets *Alarm descriptor* sont surveillés par les objets "Register monitor" de l'alarme (voir 6.2.13). L'attribut *actions* fournit le lien vers les scripts action_up et éventuellement action_down de l'objet d'envoi Push "Script table" (voir 6.2.7) qui appellent alors la méthode *push* de l'objet "Push setup" souhaité. Lorsqu'une alarme se déclenche, l'ensemble de données préalablement défini (qui peut contenir, entre autres, les objets *Alarm register* et *Alarm descriptor*) est transmis par opération Push.

Lorsque les conditions d'envoi Push sont satisfaites, les données Push sont envoyées à l'aide du service xDLMS de type de serveur/non-client non sollicité: le service DataNotification. Si les données transmises par opération Push sont longues, elles peuvent être envoyées en blocs.

Le processus d'envoi Push a lieu dans le contexte de l'application de l'AA, dans laquelle l'objet "Push setup" est visible. Le contexte de sécurité est déterminé par l'objet "Security setup" référencé depuis l'objet "Association". Les paramètres de protection des données nécessaires sont définis dans l'attribut *push_protection_parameters* qui offre les mêmes options que celles définies dans l'IC "Data Protection".

NOTE Les détails font l'objet de spécifications d'accompagnement spécifiques au projet.

Toutes les informations nécessaires au traitement des données reçues par le client doivent être:

- récupérées par le client (en lisant l'attribut *push_object_list*, par exemple);
- doivent faire partie des données transmises par opération Push;
- doivent être préalablement définies dans l'application du client.

4.4.8.2 Push setup (class_id = 40, version = 2)

4.4.8.2.1 Vue d'ensemble

L'IC "Push setup" contient une liste de références aux attributs de l'objet COSEM à transmettre par opération Push. Elle contient également la destination et la méthode d'envoi Push, ainsi que les fenêtres de temps de communication et la gestion des nouvelles tentatives.

Dans la version 1, la possibilité de protection des données a été ajoutée en offrant les mêmes options que celles définies dans l'IC "Data protection".

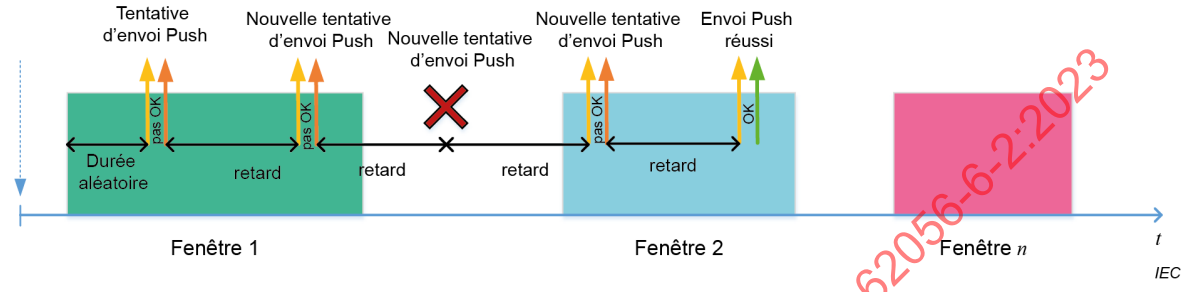
Dans la version 2, six nouvelles possibilités sont spécifiées:

- un nouveau mécanisme de sélection d'entrées permet de sélectionner les entrées associées à la dernière entrée confirmée;
- un nouveau mécanisme de sélection de colonnes permet de sélectionner explicitement les colonnes des attributs *buffer* de l'objet "Profile generic";
- l'attribut **repeat_delay** fournit maintenant une formule exponentielle qui permet d'augmenter le retard de répétition à chaque nouvelle tentative;
- un nouvel attribut **push_operation_method** permet d'indiquer si la primitive de service DataNotification.request est appelée avec Service_Class = "unconfirmed" ou "confirmed" et la façon dont l'opération de nouvelle tentative d'envoi Push fonctionne;
- un nouvel attribut **confirmation_parameters** permet de limiter l'intervalle de temps pendant lequel les entrées associées à la dernière entrée confirmée peuvent être sélectionnées;
- un nouvel attribut **last_confirmation_date_time** contient la date_time de la dernière confirmation du service DataNotification.

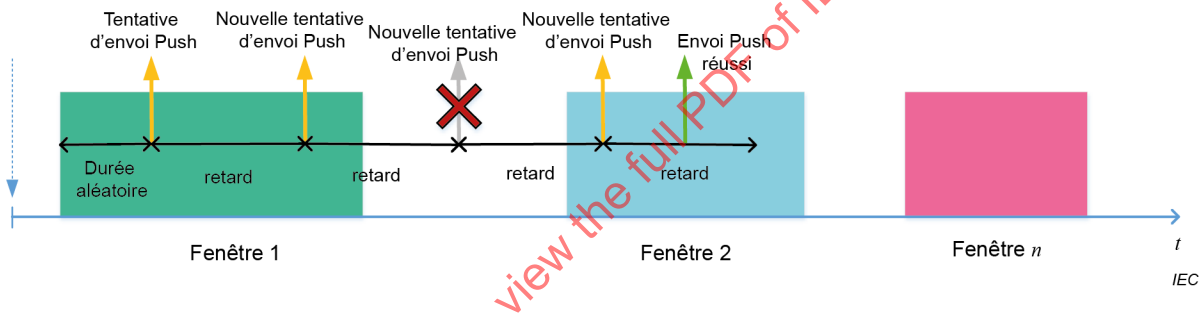
Cette version de la classe d'interfaces est destinée à faciliter l'utilisation des sélections de données relatives et absolues. Une pratique courante consisterait à utiliser une instance de l'IC "Push setup" avec une sélection de données relatives pour l'envoi Push de données de routine, et une instance de l'IC "Push setup" avec des données absolues pour des cas particuliers.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

L'envoi Push est démarré à l'appel de la méthode *push*, déclenchée par un objet d'envoi Push "Single action schedule", par un objet d'alarme "Register monitor", par un événement interne dédié ou externe. Suite au déclenchement de l'opération Push, elle est exécutée d'après les paramètres spécifiés dans l'objet "Push setup" donné. Selon les paramètres de la fenêtre de communication, l'envoi Push est exécuté immédiatement ou dès qu'une fenêtre de communication devient active, après un retard aléatoire. Si l'envoi Push n'a pas abouti, de nouvelles tentatives ont lieu. De nouvelles tentatives d'envoi Push peuvent être effectuées en cas d'indication d'échec de la couche de protocoles de support ou lorsque la confirmation est manquante. La Figure 13 représente l'opération Push avec des fenêtres, des retards et des nouvelles tentatives.



Cas a) Nouvelle tentative en cas d'échec de la couche de support



Cas b) Nouvelle tentative en cas de confirmation manquante

Figure 13 – Fenêtres d'envoi Push et retards

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Push setup		0...n	class_id = 40, version = 2			
Attribut(s)		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. push_object_list	(static)	array				x + 0x08
3. send_destination_and_method	(static)	structure				x + 0x10
4. communication_window	(static)	array				x + 0x18
5. randomisation_start_interval	(static)	long-unsigned				x + 0x20
6. number_of_retries	(static)	unsigned				x + 0x28
7. repetition_delay	(static)	structure				x + 0x30
8. port_reference	(static)	octet-string				x + 0x38
9. push_client_SAP	(static)	integer				x + 0x40
10. push_protection_parameters	(static)	array				x + 0x48
11. push_operation_method	(static)	enum				x + 0x50
12. confirmation_parameters	(static)	structure				x + 0x58
13. last_confirmation_date_time	(dyn.)	date-time				x + 0x60
Méthodes spécifiques		***o/f***				
1. push (data)		o				x + 0x38
2. reset (data)		f				x + 0x70

4.4.8.2.2 Description d'attribut

4.4.8.2.2.1 logical_name

Identifie l'instance de l'objet "Push setup". Voir 6.2.24.

4.4.8.2.2.2 push_object_list

Définit la liste des attributs à transmettre par opération Push.

À l'appel de la méthode *push* (data), les éléments sont envoyés à la destination définie dans l'attribut send_destination_and_method.

```
array        push_object_definition

push_object_definition::=     structure
{
             class_id:        long-unsigned,
             logical_name:    octet_string,
             attribute_index: integer,
             data_index:      long-unsigned,
             restriction:     restriction_element
             colonnes:        column_element
}

restriction_element::=  structure
{
      restriction_type:  enum:
                         (0) aucune,
                         (1) restriction par date,
                         (2) restriction par entrée

      restriction_value: CHOICE
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
        {
                null-data, // pas de restriction
                restriction_by_date,
                restriction_by_entry
        }
}
restriction_by_date::= structure
{
                from_date:  octet-string,
                to_date:    octet-string
}
restriction_by_entry::= structure
{
                from_entry: double-long-unsigned,
                to_entry:   double-long-unsigned
}
column_element::= array capture_object_definition
```

NOTE 1 Pour capture_object_definition, voir 4.3.6.2.3.

Où:

– class_id, logical_name et attribute_index font référence à un attribut de l'objet COSEM. Pour référencer tous les attributs d'un objet, voir IEC 62056-5-3:2023, 9.1.4.3.7, référencement attribute_0.

Si l'attribut est simple ou si l'attribut_0 est référencé, alors:

– data_index n'a aucune signification et doit être mis à 0x0000;

– restriction_type de restriction_element doit être mis à 0 (aucune) et restriction_value de restriction_element doit être "null-data";

– column_element doit être un tableau de 0 élément.

Si l'attribut est une structure ou un tableau (autre que le tampon de l'objet "Profile generic"), alors:

– data_index pointe vers un élément de la structure ou du tableau et peut prendre n'importe quelle valeur comprise entre 0x0001 et 0x0FFF;

– restriction_type de restriction_element doit être mis à 0 (aucune) et restriction_value de restriction_element doit être "null-data";

– column_element doit être un tableau de 0 élément.

Si l'attribut est un tampon de l'objet "Profile generic", alors data_index, restriction et column définissent des paramètres d'accès sélectif pour sélectionner les entrées et les colonnes.

Trois mécanismes mutuellement exclusifs sont disponibles pour sélectionner les entrées:

1) accès sélectif relatif lié aux date et heure actuelles. Les entrées des derniers intervalles de temps par rapport aux date et heure actuelles ou aux dernières entrées identifiées par data_index sont sélectionnées;

2) accès sélectif relatif lié à la dernière entrée confirmée. Les entrées pendant les intervalles de temps suivants par rapport à la dernière entrée confirmée ou aux entrées suivantes identifiées par data_index sont sélectionnées;

3) accès sélectif absolu Les entrées sur une plage de dates ou d'entrées explicitement définie, identifiées par restriction, sont sélectionnées.

Deux mécanismes mutuellement exclusifs sont disponibles pour sélectionner les colonnes:

a) un nombre défini de colonnes commençant à partir de la colonne 1 est sélectionné;

b) une liste de colonnes explicitement définie est sélectionnée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Dans le cas des mécanismes de sélection d'entrées (1) et (2), les paramètres d'accès sélectif sont définis par data_index et interprétés comme trois champs différents comme indiqué ci-dessous. Le codage est présenté dans le Tableau 21.

data_index	MS-Byte		LS-Byte
	Quartet supérieur	Quartet inférieur	

Où:

– le quarter supérieur du MS-Byte de data_index définit la manière dont les entrées sont sélectionnées;

– le quarter inférieur du MS-Byte de data_index est utilisé pour spécifier le nombre de colonnes, le cas échéant;

– LS-Byte de data_index définit le nombre d'entrées ou d'intervalles de temps.

Dans le cas des mécanismes de sélection d'entrées (1) et (2), l'élément restriction_type de restriction_element doit être à 0 et restriction_element restriction_value doit être null_data.

Dans le cas du mécanisme de sélection d'entrées (3), les paramètres d'accès sélectif sont définis par restriction_element. Dans ce cas, le quartet supérieur du MS-Byte de Data_index doit être mis à 0x0, et le LS-Byte de Data_index doit être mis à 0x00.

– pour la restriction par plage de dates, l'élément restriction_type contient (1) restriction par date et l'élément restriction_value contient la structure restriction_by_date;

– pour la restriction par entrées, l'élément restriction_type contient (2) restriction par entrée et l'élément restriction_value contient la structure restriction_by_entry;

  si l'accès sélectif n'est pas exigé, l'élément restriction_type contient (0) aucune et l'élément restriction_value contient "null-data". Ce choix doit également être opéré si l'accès sélectif relatif est utilisé.

Dans le cas du mécanisme de sélection de colonnes (a), le quartet inférieur du MS-Byte de Data_index définit le nombre de colonnes à sélectionner, à partir de la colonne 1. Dans ce cas, column_element doit être un tableau vide.

Dans le cas d'un mécanisme de sélection de colonnes (b), les colonnes à sélectionner sont identifiées par colonne_element. Dans ce cas, le quartet inférieur du MS-Byte de Data_index doit être mis à 0.

Si l'accès sélectif relatif lié à la dernière entrée confirmée est utilisé, la dernière entrée confirmée est mise à jour lorsque la primitive de service DataNotification.confim est appelée avec Result == CONFIRMED.

L'APDU DataNotification transportant les données Push est protégée comme stipulé par la suite de sécurité, la politique de sécurité et les informations de sécurité contenues dans l'objet "Security setup" référencé à partir de l'objet "Association SN" / "Association LN" dans le contexte d'application de l'AA qui est liée à l'attribut push_client_SAP, ainsi que par l'attribut push_protection_parameters. Pour les attributs auxquels aucun droit d'accès n'est accordé au sein de cette AA, ou qui sont inaccessibles pour toute autre raison, il convient de renvoyer les données nulles.

NOTE 2 Si le tableau push_object_list est vide, l'opération Push est désactivée.

NOTE 3 L'attribut push_object_list lui-même peut être transmis par opération Push afin d'identifier clairement les données transmises par opération Push.

NOTE 4 La dernière entrée confirmée est une variable interne maintenue par l'AP du serveur.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8.2.2.3 send_destination_and_method

Contient l'adresse de destination (numéro de téléphone, adresse électronique, adresse IP, par exemple) à laquelle les données spécifiées par *push_object_list* doivent être envoyées, ainsi que la méthode d'envoi.

send_destination_and_method::= structure

```
{
    transport_service:    transport_service_type,
    destination:          octet-string,
    message:              message_type
}
```

Où:

– l'élément transport_service définit le type de service utilisé pour transmettre les données par opération Push:

transport_service_type::= enum:

```
(0)          TCP;
(1)          UDP;
(2)          réservé pour FTP;
(3)          réservé pour SMTP;
(4)          SMS;
(5)          HDLC;
(6)          réservé pour M-Bus;
(7)          réservé pour ZigBee®;
(200...255)  spécifiques au fabricant.
```

– l'élément de destination contient l'adresse cible vers laquelle les données doivent être envoyées. Les éléments de l'adresse cible dépendent du service de transport utilisé.

Chaque instance d'objet "Push setup" spécifie une seule destination. Si les données sont à transmettre par opération Push vers plusieurs destinations, plusieurs objets "Push setup" doivent être instanciés;

– l'élément message_type identifie le codage de l'APDU xDLMS utilisée;

message_type::= enum:

```
(0)          APDU xDLMS codée A-XDR;
(1)          APDU xDLMS codée XML;
(128...255)  spécifiques au fabricant.
```

– toutes les autres valeurs de transport_service_type et de message_type sont réservées en vue d'une utilisation ultérieure.

4.4.8.2.2.4 communication_window

Définit les instants où les fenêtres de communication pour l'envoi Push deviennent actives (start_time) et inactives (end_time). Voir Figure 13.

```
array        window_element

window_element::= structure
{
start_time:  octet-string;
end_time:    octet-string.
}
```

start_time et end_time sont mis en forme comme indiqué en 4.1.6.1 pour *date-time*, y compris les caractères génériques.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si la fin d'une fenêtre de communication est atteinte lorsqu'un envoi Push a déjà démarré, l'opération sera menée à son terme.

Si aucune fenêtre de communication n'est définie (array [0]), l'opération Push est toujours possible.

4.4.8.2.2.5 randomisation_start_interval

Définit un retard maximal, en secondes, d'envoi de la première APDU DataNotification après appel de la méthode *push*. Ceci permet d'éviter une situation dans laquelle de nombreux serveurs peuvent pousser simultanément. Il est utilisé comme suit:

- le retard est aléatoire de 0 à la valeur maximale. Une valeur de 0 désactive le mécanisme;
- si l'envoi Push est appelé dans une fenêtre de communication active, le retard aléatoire est appliqué. Il n'est pas appliqué pour les nouvelles tentatives;
- si l'envoi Push est appelé en dehors d'une fenêtre de communication, alors le retard aléatoire est appliqué lorsque la fenêtre de communication s'ouvre;
- si le retard aléatoire s'étend au-delà de la fenêtre de communication actuelle, alors l'APDU DataNotification sera envoyée dans la fenêtre suivante si elle est disponible;
- si aucune fenêtre de communication n'est définie, le retard aléatoire est alors appliqué à chaque appel de méthode *push*, y compris toutes les nouvelles tentatives.

4.4.8.2.2.6 number_of_retries

Définit le nombre maximal de nouvelles tentatives, en cas de tentatives d'envoi Push non réussies ou ignorées. Après une opération Push réussie, aucune autre tentative d'envoi Push supplémentaire n'est réalisée tant que l'opération Push ne s'est pas de nouveau déclenchée. Pour plus de détails sur l'opération de retransmission, voir l'attribut *push_operation_method* (4.4.8.2.2.11).

4.4.8.2.2.7 repetition_delay

Définit les éléments permettant de calculer le temps de retard jusqu'à la prochaine tentative après un envoi Push infructueux. Voir Figure 13.

```
repetition_delay::= structure
{
    repetition_delay_min:          long-unsigned,
    repetition_delay_exponent:     long-unsigned,
    repetition_delay_max           long-unsigned
}
```

Où:

- repeat_delay_min est le délai minimal, en secondes, jusqu'à ce qu'une tentative d'envoi Push suivante soit lancée;
- repeat_delay_exponent est utilisé pour calculer le délai suivant;
- repeat_delay_max est le délai maximal, en secondes. Tout retard calculé sera plafonné par cette valeur.

Le retard de répétition est calculé à l'aide de la formule suivante:

$$repetition_delay = repetition_delay_min \times (repetition_delay_exponent \times 0.01)^{n-1}$$

où n désigne le nombre ordinal des nouvelles tentatives d'envoi Push avec n = 1 pour la première nouvelle tentative, etc.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE 1 Le délai de répétition lui-même n'est pas influencé par la fenêtre de communication. Toutefois, une nouvelle tentative d'envoi Push ne peut être réalisée que si une fenêtre de communication est active à cet instant. Sinon elle est traitée comme une tentative d'envoi Push infructueuse.

NOTE 2 Les données Push ne sont pas enregistrées dans un tampon intermédiaire. En cas de nouvelles tentatives, les valeurs en cours des attributs peuvent changer à chaque nouvelle tentative d'envoi Push.

4.4.8.2.2.8 port_reference

Contient le nom logique d'un objet de configuration de port de communication permettant la sélection d'un canal de communication spécifique pour l'envoi Push sur la base du transport_service_type. Ceci s'applique principalement dans les cas où plusieurs canaux du même type sont pris en charge.

Si cette information n'est pas disponible ou nécessaire, l'attribut peut être laissé vide (octet-string [0]).

4.4.8.2.2.9 push_client_SAP

Définit le SAP du client vers lequel l'envoi Push est dirigé dans la couche de support du service DataNotification. Le processus d'envoi Push a lieu dans le contexte de l'application de l'AA qui est liée à l'attribut push_client_SAP. Le contexte de sécurité est déterminé par l'objet "Security setup" référencé dans l'objet "Association SN / Association LN" rattaché.

4.4.8.2.2.10 push_protection_parameters

Spécifie tous les paramètres de protection à appliquer à l'APDU DataNotification lorsque les données Push sont envoyées. Il offre les mêmes options que l'IC "Data Protection", mais il est lié à l'envoi des données définies dans l'attribut push_object_list.

array protection_parameters_element

```
protection_parameters_element::= structure
{
        protection_type: enum:
                (0) authentification,
                (1) cryptage,
                (2) authentification et cryptage,
                (3) signature numérique

        protection_options: structure
        {
                transaction_id:             octet-string,
                originator_system_title:    octet-string,
                recipient_system_title:     octet-string,
                other_information:          octet-string,

                key_info:   key_info_element
        }
}
```

Où:

– transaction_id contient l'identifiant de la transaction;

– originator_system_title contient le titre système de l'émetteur qui applique la protection;

– recipient_system_title contient le titre de système du destinataire qui va vérifier et retirer la protection donnée;

– other_information contient d'autres informations. Son contenu peut être précisé dans des spécifications d'accompagnement spécifiques au projet. Une chaîne d'octets de longueur 0 indique que ce champ n'est pas utilisé;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– key_info contient les informations dont a besoin le destinataire pour obtenir la bonne clé pour vérifier et retirer l'authentification et le cryptage. Dans le cas de la signature numérique, key_info n'est pas nécessaire et doit être une structure de 0 élément.

Les champs transaction-id ….other-information sont des CHAÎNES D'OCTETS codées A-XDR. Le cas échéant, la longueur et la valeur de chaque champ sont incluses dans l'AAD.

```
key_info_element::= structure
{
key_info_type: enum:
        (0) identified_key,
        -- utilisé avec identified_key_info_options

            (1) wrapped_key,
            -- utilisé avec wrapped_key_info_options

            (2) agreed_key,
            -- utilisé avec agreed_key_info_options

key_info_options: CHOICE
    {
        identified_key_info_options,
        wrapped_key_info_options,
        agreed_key_info_options
    }
}
identified_key_info_options::= enum:
            (0) global_unicast_encryption_key,
            (1) global_broadcast_encryption_key

wrapped_key_info_options::= structure
{
kek_id:     enum:
            (0) master_key
key_ciphered_data: octet-string
}
agreed_key_info_options::= structure
{
key_parameters: octet-string,
key_ciphered_data: octet-string
}
```

Cet attribut est écrit en premier par le client. Le serveur peut avoir besoin de renseigner certains éléments supplémentaires.

L'utilisation des différents éléments est la même que celle spécifiée dans le Tableau 22 et le Tableau 23 de l'IC "Data protection" (class_id = 30, version = 0).

4.4.8.2.2.11 push_operation_method

Définit si la primitive de service DataNotification.request est appelée avec Service_Class == Unconfirmed ou Confirmed et la façon dont la nouvelle tentative d'envoi Push fonctionne.

enum:

(0) non confirmé, nouvelle tentative sur échec de la couche de protocoles de support;
(1) non confirmé, nouvelle tentative sur confirmation de couche de protocoles de support manquante;
(2) confirmé, nouvelle tentative sur confirmation manquante.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Dans le cas (0), le délai de répétition pour la nouvelle tentative suivante est démarré lors de l'échec de la couche de support déclaré par la primitive de service DataNotification.confirm.

Dans les cas (1) et (2), le délai de répétition pour la nouvelle tentative suivante est démarré lorsque la primitive de service DataNotification.request est appelée.

Dans les cas (0) et (1), l'opération Push est considérée comme réussie lorsque la confirmation de la couche de support est signalée en appelant la primitive de service DataNotification.confim avec Result == CONFIRMED.

Dans le cas (2), l'opération Push est considérée comme réussie lorsque l'AL du serveur reçoit l'APDU data-notification-confirm et appelle la primitive de service DataNotification.confirm avec Service_Class == Confirmed et Result == CONFIRMED.

4.4.8.2.2.12 confirmation_parameters

Définit la sélection des entrées définies par data_index pour éviter de pousser les données trop loin dans le passé. Si les entrées n'ont pas encore été confirmées, alors toutes les entrées sont sélectionnées. Cet attribut s'applique uniquement à l'accès sélectif relatif lié à la dernière confirmation.

```
confirmation_parameters::= structure
{
        confirmation_start_date:    date-time,
        confirmation_interval:      double-long-unsigned
}
```

Où:

- confirmation_start_date est la date et heure de début de sélection des entrées. Les champs de *date-time* non spécifiés ne sont pas utilisés. Si tous les champs ne sont pas spécifiés, l'utilisation de confirmation_start_date est désactivée;
- confirmation_interval est un intervalle de temps, en secondes, par rapport aux date et heure actuelles et en remontant dans le temps. Si confirmation_interval est mis à zéro, l'utilisation de confirmation_interval est désactivée.

4.4.8.2.2.13 last_confirmation_date_time

Contient la date et l'heure où l'AL a appelé le plus récemment la primitive de service DataNotification.confirm avec Result == CONFIRMED.

date-time est mis en forme comme spécifié en 4.1.6.1.

4.4.8.2.3 Description de la méthode

4.4.8.2.3.1 push (data)

Active le processus d'envoi Push conduisant à une tentative d'envoi d'une APDU DataNotification transportant les données Push.

```
data::= integer(0)
```

4.4.8.2.3.2 reset (data)

Remet le processus d'envoi Push à l'état initial.

```
data::= integer(0)
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.8.2.4 Codage des paramètres d'accès sélectif avec data_index

Le Tableau 21 donne des détails sur la façon dont les paramètres d'accès sélectif sont définis à l'aide de data_index.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 21 – Codage des paramètres d'accès sélectif avec data_index

	Quartet supérieur du MS_Byte: sélectionne les intervalles de temps ou les entrées.
0xF	**Dernier nombre de mois complets:** Accès sélectif au *tampon*, donnant toutes les entrées du dernier nombre de mois complets et la première entrée à minuit du mois en cours.
0xE	**Dernier nombre de jours complets:** Accès sélectif au *tampon* du profil, donnant toutes les entrées du dernier nombre de jours complets et la première entrée à minuit le jour même.
0xD	**Dernier nombre d'heures complètes:** Accès sélectif au *tampon*, donnant toutes les entrées du dernier nombre d'heures complètes et la première entrée de l'heure en cours.
0xC	**Dernier nombre de minutes complètes:** Accès sélectif au *tampon*, donnant toutes les entrées du dernier nombre de minutes complètes et la première entrée de la minute en cours.
0xB	**Dernier nombre de secondes:** Accès sélectif au *tampon*, donnant lieu à toutes les entrées du dernier nombre de secondes.
0xA	**Dernier nombre de mois complets, y compris le mois en cours:** Identique à 0xF ci-dessus, sauf que toutes les entrées sont désormais récupérées.
0x9	**Dernier nombre de jours complets, y compris le jour en cours:** Identique à 0xE ci-dessus, sauf que toutes les entrées sont désormais récupérées.
0x8	**Dernier nombre d'heures complètes, y compris l'heure en cours:** Identique à 0xD ci-dessus, sauf que toutes les entrées sont désormais récupérées.
0x7	**Dernier nombre de minutes complètes, y compris la minute en cours:** Identique à 0xC ci-dessus, sauf que toutes les entrées sont désormais récupérées.
0x6…0x2	Les valeurs 0x6 à 0x2 s'appliquent uniquement à la classe d'interfaces "Push setup" version 2 (voir 4.4.8.2).
0x6	**Nombre de mois complets après la dernière confirmation:** Équivaut à un accès sélectif au *tampon* de profil donnant toutes les entrées du nombre de mois complets après la dernière entrée confirmée.
0x5	**Nombre de jours complets après la dernière confirmation:** Équivaut à un accès sélectif au *tampon* de profil donnant toutes les entrées du nombre de jours complets après la dernière entrée confirmée.
0x4	**Nombre d'heures complètes après la dernière confirmation:** Équivaut à un accès sélectif au *tampon* de profil donnant toutes les entrées du nombre d'heures complètes après la dernière entrée confirmée.
0x3	**Nombre complet de minutes après la dernière confirmation:** Équivaut à un accès sélectif au *tampon* de profil donnant toutes les entrées du nombre de minutes complètes après la dernière entrée confirmée.
0x2	**Nombre d'entrées après la dernière confirmation**: Équivaut à un accès sélectif au *tampon* de profil donnant le nombre d'entrées après la dernière entrée confirmée.
0x1	**Dernier nombre d'entrées**
0x0	Un **attribut complet** ou un seul élément d'un attribut avec un type de données complexes est sélectionné (quartet inférieur du MS-Byte 0x0…0xF, LS-Byte 0x00…0xFF).
	Quartet inférieur du MS-Byte !=0: Définit le nombre de colonnes d'un *tampon* "Profile generic" sélectionné.
0x0	Toutes les colonnes
0x1 à 0xF	Nombre de colonnes, en commençant par la colonne 1.
0x00…0xFF	LS-Byte: – lorsque les entrées dans les intervalles de temps sont sélectionnées, spécifie le nombre d'intervalles de temps, la valeur 0x00 définissant toutes les périodes; – lorsque les entrées sont sélectionnées, spécifie le nombre d'entrées, la valeur 0x00 définissant toutes les entrées.
Exemple 1	0xE401: les entrées du dernier jour complet sont sélectionnées. Les 4 premières colonnes sont incluses.
Exemple 2	0xA300: les entrées pour zéro dernier mois complet et le mois en cours sont sélectionnées. Les 3 premières colonnes sont incluses.
Exemple 3	0x800C: les entrées des 12 dernières heures complètes sont sélectionnées. Toutes les colonnes sont incluses.
Exemple 4	0x1080: les 128 dernières entrées sont sélectionnées. Toutes les colonnes sont incluses.
Exemple 5	0x4608: Les entrées des 8 dernières heures complètes après la dernière entrée confirmée sont sélectionnées. Les 6 premières colonnes sont incluses.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9 COSEM data protection (class_id = 30, version = 0)

4.4.9.1 Vue d'ensemble

Les instances de cette IC permettent d'appliquer une protection cryptographique sur les données COSEM, c'est-à-dire sur les valeurs d'attribut et sur les paramètres d'appel et de retour de la méthode. Pour ce faire, il suffit d'accéder indirectement aux attributs et/ou aux méthodes d'autres objets COSEM par l'intermédiaire de la classe d'interfaces "Data protection" qui fournit les mécanismes et les paramètres nécessaires à l'application/la vérification/la suppression/la protection des données COSEM.

NOTE 1 L'accès inclut la lecture/l'écriture/la capture/l'envoi Push des attributs d'objet COSEM ou des méthodes d'appel.

NOTE 2 Si les attributs et les méthodes des objets COSEM sont directement accessibles, la protection peut être assurée en protégeant les APDU xDLMS conformément à la politique de sécurité et aux droits d'accès applicables.

NOTE 3 Pour obtenir des définitions et des abréviations liées à la sécurité cryptée, voir IEC 62056-5-3:2023, Article 3.

La protection des données COSEM est conforme et complète la protection des APDU xDLMS telle que définie dans l'IEC 62056-5-3:2023, 5.7.

Les cas d'utilisation de la protection des données COSEM incluent, mais sans s'y limiter:

- la lecture ou l'écriture d'un ensemble prédéfini de valeurs d'attribut protégées;
- le stockage d'un ensemble prédéfini de valeurs d'attribut protégées dans les objets "Profile generic" en vue d'une récupération ultérieure;
- l'envoi Push d'un ensemble prédéfini de valeurs d'attribut protégées;
- la lecture ou l'écriture des attributs sélectionnés d'autres objets COSEM avec protection;
- l'appel d'une méthode d'un autre objet COSEM avec les paramètres protégés d'appel et de retour de la méthode.

La protection peut combiner authentification/cryptage/signature numérique et peut être appliquée sous forme de couches. Les parties qui appliquent et retirent la protection sont le serveur DLMS®/COSEM et une autre partie identifiée, qui peut être un client DLMS®/COSEM ou un tiers.

L'application de la protection des données entre un serveur DLMS®/COSEM et un tiers permet d'assurer la confidentialité des données critiques/sensibles auprès du client par lequel le tiers accède au serveur. La signature des données COSEM par un tiers permet de prendre en charge la non-répudiation.

Pour une protection de bout en bout entre les tiers et les serveurs, voir aussi IEC 62056-5-3:2023, 4.1.7 et 5.2.5.

Les paramètres de protection sont toujours contrôlés par le client, certains éléments étant renseignés par le serveur, selon le cas.

La suite de sécurité est déterminée par l'objet "Security setup" référencé depuis l'objet "Association SN"/"Association LN".

La Figure 14 représente le modèle COSEM de protection des données et la relation entre un objet "Data protection" et d'autres objets COSEM.

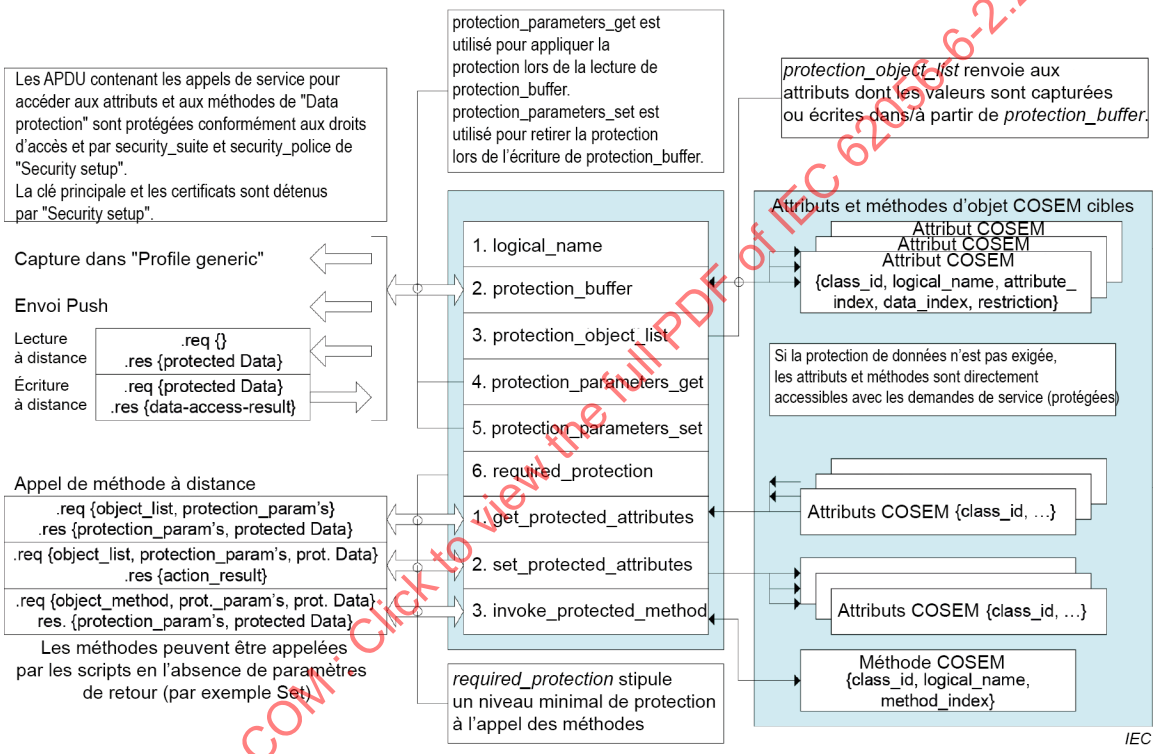
IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Deux mécanismes permettent d'accéder aux attributs d'autres objets COSEM avec des données protégées:

- la lecture ou l'écriture de l'attribut *protection_buffer*. L'attribut *protection_buffer* peut également être capturé dans les objets "Profile generic" ou être transmis par opération Push à l'aide des objets "Push setup";
- l'appel de la méthode *get_protected_attributes/set_protected_attributes*.

La méthode *invoke_protected_method* permet d'accéder à une méthode d'un autre objet COSEM avec des données protégées.

Les APDU contenant des appels de service pour accéder aux attributs et aux méthodes des objets "Data protection" sont protégées conformément aux droits d'accès à ces attributs et méthodes, et par *security_suite* et *security_policy* de l'objet "Security setup".



La clé principale et les certificats (exigés par la suite de sécurité) sont détenus par "Security setup".

Figure 14 – Modèle COSEM de protection des données

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

La protection des données COSEM est appliquée et retirée dans les différents cas ci-dessous:

a) lorsque l'attribut *protection_buffer* est lu/capturé dans un objet "Profile generic"/transmis par opération Push:
   - les attributs déterminés par *protection_object_list* sont capturés;
   - la protection selon *protection_parameters_get* est appliquée sur l'ensemble des attributs, et le résultat est placé dans *protection_buffer*;
   - la valeur de *protection_buffer* est renvoyée/capturée dans le *buffer* de "Profile generic"/transmise par opération Push à l'aide des objets "Push setup";

b) lorsque *protection_buffer* est écrit:
   - les données protégées sont écrites dans *protection_buffer*;
   - la protection selon *protection_parameters_set* est retirée et les valeurs d'attribut obtenues sont écrites dans les attributs spécifiés par *protection_object_list*;

c) lorsque la méthode *get_*protected*_attributes* est appelée:

- les attributs déterminés par l'élément object_list de get_protected_attributes_request sont capturés;
- la protection selon l'attribut *required_protection* et la réponse protection_parameters est appliquée. Si protection_parameters ne satisfait pas à required_protection, l'appel de la méthode n'aboutit pas;
- les valeurs d'attribut protégées sont renvoyées;

d) Lorsque la méthode *set_protected_attributes* est appelée:

- la protection de protected_attributes est vérifiée et retirée à l'aide des protection_parameters qui doivent satisfaire à *required_protection*;
- les valeurs d'attribut obtenues sont placées dans les attributs spécifiés par l'élément object_list;

e) Lorsque la méthode *invoke_protected_method* est appelée:

- la protection par les paramètres protégés d'appel de la méthode est retirée à l'aide des paramètres de protection de la requête qui doivent satisfaire à *required_protection*;
- la méthode spécifiée par l'élément object_method d'invoke_protected_method_request est appelée avec ce paramètre d'appel de la méthode;
- sur les paramètres de retour, la protection à l'aide des paramètres de protection de réponse qui doivent satisfaire à *required_protection* s'applique. Si protection_parameters ne satisfait pas à required_protection, l'appel de la méthode n'aboutit pas;
- les paramètres protégés de retour de la méthode sont renvoyés.

La Figure 15 donne un exemple de construction de données protégées dans *protection_buffer* en fonction des attributs déterminés par *protection_object_list* selon *protection_parameters_get*. Voir aussi IEC 62056-5-3:2023, Figure 29.

Les étapes suivantes sont mises en œuvre lors de la lecture de l'attribut *protection_buffer*:

f) conditions préalables: *protection_object_list*, *protection_parameters_get*, clé principale, agrément de clé et certificats de signature numérique, selon le cas;

g) capturer les attributs d'objet COSEM déterminés par *protection_object_list* et créer Data, une structure contenant les données individuelles des attributs capturés;

h) protéger Data selon *protection_parameters_get*;

NOTE 4 Dans l'exemple de la Figure 15, deux couches de protection sont appliquées:

– la première couche est une combinaison de compression/de cryptage/d'authentification, comme déterminée par le SC d'octet de contrôle de sécurité, ce qui donne (C)Data;

– la seconde couche est la signature numérique appliquée à (C)Data;

i) placer les données protégées, de type octet-string, dans *protection_buffer*;

j) renvoyer la valeur de *protection_buffer*.

Il peut s'avérer nécessaire de lire également protection_parameters_get pour obtenir les paramètres de protection permettant au destinataire de vérifier/retirer la protection.

Le compteur d'appels utilisé dans le cadre de la protection est appliqué/retiré en fonction de la clé utilisée. Lorsque la protection est appliquée, le compteur d'appels correspondant est incrémenté. Si la clé est modifiée, le compteur d'appels doit être remis à 0.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

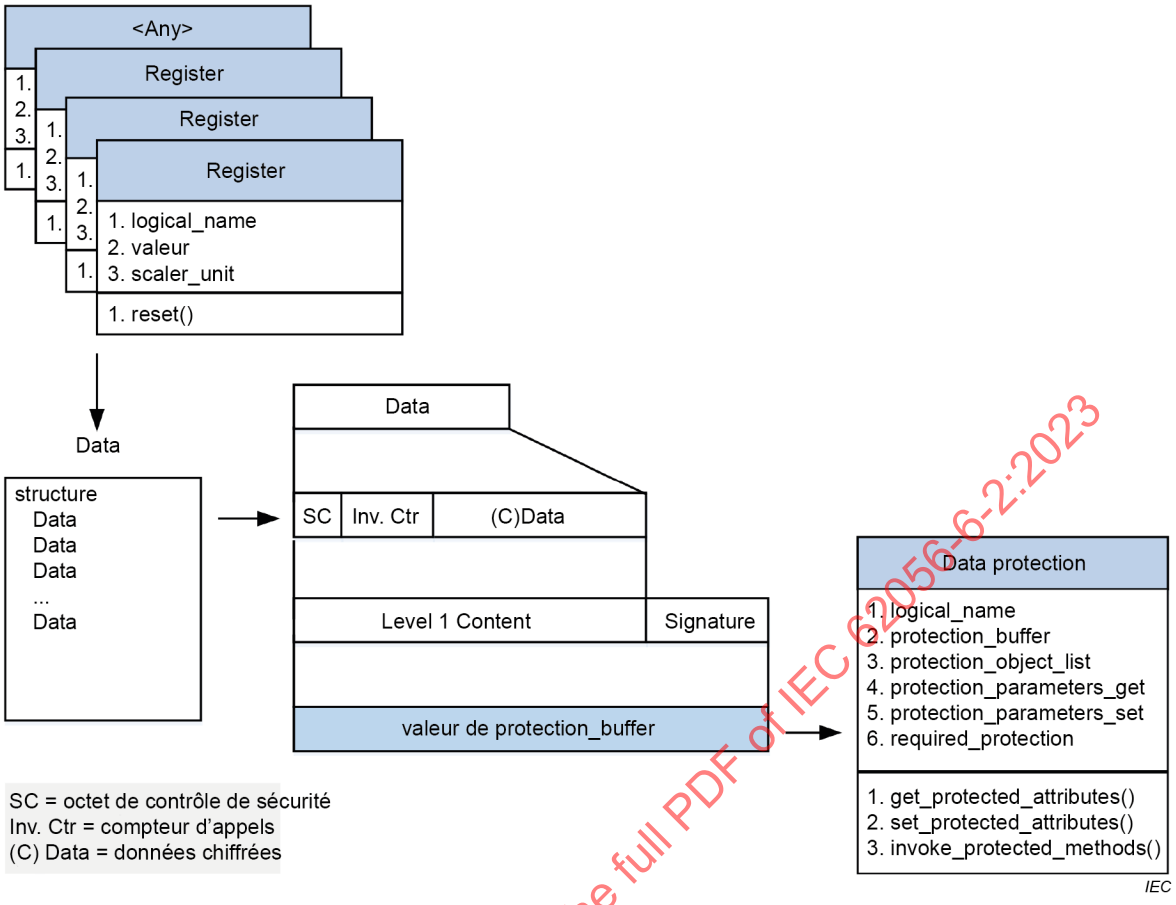


Figure 15 – Exemple: lecture de l'attribut *protection_buffer*

Data protection		**0...n**	**class_id = 30, version = 0**			
Attribut(s)		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. protection_buffer	(dyn.)	octet-string				x + 0x08
3. protection_object_list	(static)	array				x + 0x10
4. protection_parameters_get	(static)	array				x + 0x18
5. protection_parameters_set	(static)	array				x + 0x20
6. required_protection	(static)	enum				x + 0x28
Méthodes spécifiques		***o/f***				
1. get_protected_attributes (data)		o				x + 0x30
2. set_protected_attributes (data)		o				x + 0x38
3. invoke_protected_method (data)		o				x + 0x40

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9.2 Description d'attribut

4.4.9.2.1 logical_name

Identifie l'instance de l'objet "Data protection". Voir 6.2.36.

4.4.9.2.2 protection_buffer

Contient les données protégées.

Lorsqu'il est lu, les attributs déterminés par *protection_object_list* sont capturés, puis la protection est appliquée selon *protection_parameters_get*.

Lorsqu'il est écrit, les données protégées sont placées dans *protection_buffer*, puis la protection est vérifiée/retirée selon *protection_parameters_set*, et les attributs déterminés par *protection_object_list* sont définis.

4.4.9.2.3 protection_object_list

Définit la liste des attributs à capturer dans *protection_buffer* lorsqu'il est lu ou la liste des attributs à définir lorsque *protection_buffer* est écrit.

Deux mécanismes d'accès sélectif mutuellement exclusifs sont disponibles:

– l'accès sélectif relatif, c'est-à-dire que les entrées définies selon la date ou l'entrée en cours sont renvoyées: ce mécanisme est commandé par l'élément data_index;

– l'accès sélectif absolu, c'est-à-dire que les entrées situées dans une plage de dates ou d'entrées explicitement définie sont renvoyées: ce mécanisme est commandé par l'élément restriction.

array object_definition

object_definition::= structure
{
class_id: long-unsigned,
logical_name: octet-string,
attribute_index: integer,
data_index: long-unsigned,
restriction: restriction_element
}

Où:

– attribute_index est un pointeur vers l'attribut au sein de l'objet, identifié par class_id et logical_name: attribute_index 1 fait référence au 1er attribut (c'est-à-dire *logical_name*), attribute_index 2 au 2e attribut, etc.; attribute_index 0 fait référence à tous les attributs publics;

– data_index est un pointeur qui sélectionne un ou plusieurs éléments particuliers d'un attribut avec un type de données complexe (structure ou tableau):

- si le type de données de l'attribut est simple, alors data_index n'a aucune signification;
- si le type de données de l'attribut est une structure ou un tableau, data_index pointe vers un ou plusieurs éléments spécifiques de la structure ou du tableau;
- lorsque l'attribut est le *buffer* d'un objet "Profile generic", data_index contient des paramètres d'accès sélectif en fonction de la date ou de l'entrée en cours.

data_index:	MS-Byte		LS-Byte
	Quartet supérieur	Quartet inférieur	

– 0x0000 = identifie tout l'attribut;

– 0x0001 à 0x0FFF = identifie un élément dans l'attribut complexe. Le premier élément de l'attribut complexe est identifié par data_index 1;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– 0x1000 à 0xFFFF = accès sélectif au tableau contenant le buffer d'un objet "Profile generic". Le data-index sélectionne les entrées sur un certain nombre de périodes (récentes) ou un certain nombre d'entrées (récentes), ainsi que les colonnes du tableau.

Le codage est spécifié dans le Tableau 25.

Lorsque l'attribut est le *buffer* d'un objet "Profile generic", restriction_element spécifie des paramètres d'accès sélectif dans une plage de dates ou d'entrées définie de manière explicite.

```
restriction_element::=   structure
{
restriction_type:  enum:
                   (0)   aucune,
                   (1)   restriction par date,
                   (2)   restriction par entrée
restriction_value: CHOICE
{
      null-data,                  // pas de restriction
      restriction_by_date,
      restriction_by_entry
}
}

restriction_by_date::=   structure
{
from_date:         octet-string,
to_date:           octet-string
}

restriction_by_entry::=  structure
{
from_entry:        double-long-unsigned,
to_entry:          double-long-unsigned
}
```

– restriction_element définit l'accès sélectif absolu au buffer "Profile generic" par plage de dates (from_date à to_date) ou par entrées (from_entry à to_entry). Pour utiliser ce mécanisme d'accès sélectif absolu, data_index doit être:

  - positionné à 0x0 pour le quartet supérieur de l'octet MS;
  - positionné de 0x0 à 0xF pour le quartet inférieur de l'octet MS, conformément au Tableau 21;
  - positionné à 0x00 pour le quartet supérieur de l'octet inférieur;

– restriction_element est composé de restriction_type et de restriction_value:

  - pour la restriction par plage de dates, l'élément restriction_type contient (1) (restriction par date) et l'élément restriction_value contient la structure restriction_by_date;
  - pour la restriction par entrées, l'élément restriction_type contient (2) (restriction par entrée) et l'élément restriction_value contient la structure restriction_by_entry;
  - sinon, l'élément restriction_type contient (0) (aucune) et l'élément restriction_value contient null-data. Ce choix doit également être effectué si l'accès sélectif relatif doit être utilisé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9.2.4 protection_parameters_get

Contient tous les paramètres nécessaires à la spécification de la protection appliquée lorsque *protection_buffer* est lu.

```
array protection_parameters_element

protection_parameters_element::= structure

{
protection_type:  enum:
authentification
cryptage,
authentification et cryptage,
signature numérique
protection_options:  structure
{
        transaction_id:                 octet-string,
        originator_system_title:        octet-string,
        recipient_system_title:         octet-string,
        other_information:              octet-string,
        key_info:                       key_info_element
}
}
```

Où:

– transaction_id contient l'identifiant de la transaction;

– originator_system_title contient le titre de système de l'émetteur qui applique la protection;

– recipient_system_title contient le titre de système du destinataire qui va vérifier et retirer la protection donnée;

– other_information contient d'autres informations. Son contenu peut être précisé dans des spécifications d'accompagnement spécifiques au projet. Une chaîne d'octets de longueur 0 indique que ce champ n'est pas utilisé;

– key_info contient les informations dont a besoin le destinataire pour obtenir la bonne clé pour vérifier et retirer l'authentification et le cryptage. Dans le cas de la signature numérique, key_info n'est pas nécessaire et doit être une structure de 0 élément.

Les champs transaction-id ….other-information sont des CHAÎNES D'OCTETS codées A-XDR. Le cas échéant, la longueur et la valeur de chaque champ sont incluses dans l'AAD.

```
key_info_element::= structure
{
key_info_type:    enum:
                  (0) identified_key,
                  -- utilisé avec identified_key_info_options
            (1) wrapped_key,
                  -- utilisé avec wrapped_key_info_options
            (2) agreed_key
                  -- utilisé avec agreed_key_info_options
key_info_options: CHOICE
{
      identified_key_info_options,
      wrapped_key_info_options,
      agreed_key_info_options
}
}
identified_key_info_options::=      enum:
global_unicast_encryption_key,
global_broadcast_encryption_key
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
wrapped_key_info_options::=structure
{
kek_id:                    enum:
                (0) master_key,

key_ciphered_data:     octet-string
}
agreed_key_info_options::=   structure
{
key_parameters:        octet-string,
key_ciphered_data:     octet-string
}
```

Cet attribut est écrit en premier par le client. Le serveur peut renseigner certains éléments supplémentaires, auquel cas cet attribut doit être relu par le client (et, le cas échéant, transféré au tiers) de manière à pouvoir utiliser ces paramètres pour vérifier/retirer la protection.

Pour l'utilisation des différents éléments, voir Tableau 22 et Tableau 23.

4.4.9.2.5 protection_parameters_set

Contient tous les paramètres nécessaires à la vérification/au retrait de la protection appliquée lorsque *protection_buffer* est écrit.

```
array protection_parameters_element
```

protection_parameters_element: voir l'attribut *protection_parameters_get* (4.4.9.2.4).

Cet attribut est écrit par le client et utilisé par le serveur pour vérifier et retirer la protection.

Pour l'utilisation des différents éléments, voir Tableau 22 et Tableau 24.

4.4.9.2.6 required_protection

Stipule la protection exigée des valeurs d'attribut/paramètres d'appel et de retour des méthodes accessibles par l'intermédiaire de l'objet "Data protection".

```
enum:
```

Si la valeur enum est interprétée comme un unsigned, la signification de chaque bit est la suivante:

Bit	Protection exigée
0	non utilisé, doit être positionné à 0;
1	non utilisé, doit être positionné à 0;
2	demande authentifiée;
3	demande cryptée;
4	demande signée numériquement;
5	réponse authentifiée;
6	réponse cryptée;
7	réponse signée numériquement.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.9.3 Description de la méthode

4.4.9.3.1 get_protected_attributes (data)

Récupère la valeur des attributs spécifiés par l'élément object_list, la protection selon protection_parameters de get_protected_ attributes_response étant appliquée.

Paramètres d'appel de la méthode:

```
data::= get_protected_attributes_request

get_protected_attributes_request::= structure
{
object_list:                array object_definition,
protection_parameters:      array protection_parameters_element,
}
```

Paramètres de retour:

```
data::= get_protected_attributes_response

get_protected_attributes_response::= structure
{
protection_parameters: array protection_parameters_element,
protected_attributes:       octet-string
}
```

Où:

– object_list: voir l'attribut *protection_object_list* (4.4.9.2.3);

– protection_parameters::= array protection_parameters_element.

protection_parameters_element: voir l'attribut *protection_parameters_get* (4.4.9.2.4).

Les protection_parameters de get_protected_attributes_response sont élaborés en copiant les protection_parameters depuis get_protected_attributes_request, sauf que le serveur peut souhaiter renseigner certains éléments. La partie distante utilise les protection_parameters de get_protected_attributes_response pour vérifier/retirer la protection des protected_attributes renvoyés.

Pour l'utilisation des différents éléments, voir Tableau 22 et Tableau 25.

4.4.9.3.2 set_protected_attributes (data)

Définit la valeur des attributs spécifiés par l'élément object_list, après avoir vérifié et retiré la protection de l'élément protected_attributes conformément à l'élément protection_parameters.

Paramètres d'appel de la méthode:

```
data::= set_protected_attributes_request
set_protected_attributes_request::= structure
{
object_list:                    array object_definition,
protection_parameters:          array protection_parameters_element,
protected_attributes:           octet-string
}
```

Où:

– object_list: voir l'attribut *protection_object_list* (4.4.9.2.3);

– protection_parameters::= array protection_parameters_element.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

protection_parameters_element: voir l'attribut *protection_parameters_get* (4.4.9.2.4).

Pour l'utilisation des différents éléments, voir Tableau 22 et Tableau 26.

4.4.9.3.3 invoke_protected_method (data)

Appelle la méthode spécifiée par l'élément object_method, après avoir vérifié et retiré la protection des protected_method_invocation_parameters selon protection_parameters de invoke_protected_method_ request.

À la suite de l'appel de la méthode spécifiée par l'élément object_method, la protection selon protection_parameters d'invoke_protected_ method_response (qui doit satisfaire à *required_protection*) s'applique aux paramètres de retour, et invoke_protected_method_response composé de protection_parameters et de protected_method_return_parameters est renvoyé.

Paramètres d'appel de la méthode:

data::= invoke_protected_method_request

```
invoke_protected_method_request::=     structure
{
object_method:                         object_method_definition,

protection_parameters: array           protection_parameters_element,
protected_method_invocation_parameters:        octet-string
}

object_method_definition::=  structure
{
class_id:          long-unsigned,
logical_name:      octet_string,
method_index:      integer,
}
```

Paramètres de retour:
data::= invoke_protected_method_response

```
invoke_protected_method_response::= structure
{
protection_parameters: array           protection_parameters_element,
protected_method_return_parameters:    octet-string
}
```

protection_parameters_element: voir la spécification de l'attribut protection_parameters_get (4.4.9.2.4).

Si la méthode appelée ne fournit pas de paramètres de retour, invoke_protected_method_response doit être une structure, protection_parameters étant un tableau de zéro élément, et protected_method_return_parameters étant une chaîne d'octets de longueur nulle.

Pour vérifier et supprimer la protection des protected_method_invocation_parameters, le serveur utilise les paramètres de protection dans invoke_protected_method_ request qui doit satisfaire à *required_protection*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Les protection_parameters de l'invoke_protected_method_response sont élaborés en copiant les protection_parameters d'invoke_protected_ method_request dans invoke_protected_method_response, sauf que le serveur peut souhaiter renseigner certains éléments.

Les protection_parameters d'invoke_protected_method_response (qui doivent satisfaire à *required_protection*) sont ensuite appliqués pour protéger les données dans protected_method_return_parameters.

4.4.9.4 Key_information et paramètres de protection

Pour l'utilisation des différents éléments, voir Tableau 22 et Tableau 27.

Tableau 22 – Informations relatives aux clés exigées pour établir les clés de protection des données

Choix des informations relatives aux clés		Commentaire
key_info_options		
(0) **identified_key_info_options**	S	L'EK est identifiée
(0) global_unicast_encryption_key	S	GUEK
(1) global_broadcast_encryption_key	S	GBEK
(1) **wrapped_key_info_options**	S	L'EK est transportée à l'aide d'un enveloppement de clé
kek_id	O	
(0) master_key	O	Identifie la clé utilisée pour envelopper key_ciphered_data. 0 = clé principale (KEK)
key_ciphered_data	O	Clé générée de manière aléatoire enveloppée avec KEK.
(2) **agreed_key_info_options**	S	La clé fait l'objet d'un accord entre les parties en utilisant: – le schéma One-Pass Diffie-Hellman C(1e, 1s, ECC CDH), ou – le schéma Static Unified Model C(0e, 2s, ECC CDH).
key_parameters	O	Identificateur du mécanisme d'agrément de clé: 0x01: C(1e, 1s, ECC CDH) 0x02: C(0e, 2s, ECC CDH) Tous les autres sont réservés.
key_ciphered_data	O	– Dans le cas du schéma C(1e, 1s, ECC CDH): la clé publique Qu de la paire de clés d'agrément de clé éphémère de la partie U, signée avec la clé de signature numérique privée de la partie U. – En cas de schéma C(0e, 2s, ECC CDH): chaîne d'octets de longueur nulle. NOTE Dans le second cas, la partie U doit fournir une valeur unique, NonceU. Voir IEC 62056-5-3:2023, 5.3.4.6.4 et 5.3.4.6.5.
O: Obligatoire (partie d'une structure) S: Sélectionnable (partie d'un CHOICE)		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 23 – Paramètres de protection de l'attribut *protection_parameters_get*

protection_parameters_get		Écrit par le client	Renseigné par le serveur
array protection_parameters_element	O	x	
protection_type	O	x	
(0) authentification	S	x	
(1) cryptage	S	x	
(2) authentification et cryptage	S	x	
(3) signature numérique	S	x	
protection_options	O	x	
transaction_id	O	x	
originator_system_title (doit contenir le titre de système du serveur)	O	x	
recipient_system_title (doit contenir le titre de système de la partie distante)	O	x	
other_information	O	x	
key_info	O	x	
key_info_type	O	x	
(0) identified_key	S	x	
(1) wrapped_key	S	x	
(2) agreed_key	S	x	
key_info_options	O	x	
identified_key_options	S	x	
(0) global_unicast_encryption_key	S	x	
(1) global_broadcast_encryption_key	S	x	
wrapped_key_options	S	x	
kek_id	O	x	
(0) master_key	O	x	
key_ciphered_data	O	x	
agreed_key_options	S	x	
key_parameters	O	x	
key_ciphered_data	O		x
L'attribut *protection_parameters_get* est écrit par le client, mais si key_info_type est (2) est la clé convenue, l'attribut key_ciphered_data d'agreed_key_options est vide. Si le schéma One-pass Diffie-Hellman c(1e, 1s, ECC CDH) est utilisé, cet élément est renseigné par le serveur, et la partie distante doit relire *protection_parameters_get* afin d'être en mesure de vérifier et de retirer la protection.			
O: Obligatoire (partie d'une structure). S: Sélectionnable (partie d'un CHOICE).			

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 24 – Paramètres de protection de l'attribut *protection_parameters_set*

protection_parameters_set		Écrit par le client	Renseigné par le serveur
array protection_parameters_element	O	x	
protection_type	O	x	
(0) authentification	S	x	
(1) cryptage	S	x	
(2) authentification et cryptage	S	x	
(3) signature numérique	S	x	
protection_options	O	x	
transaction_id	O	x	
originator_system_title (doit contenir le titre de système de la partie distante)	O	x	
recipient_system_title (doit contenir le titre de système du serveur)	O	x	
other_information	O	x	
key_info	O	x	
key_info_type	O	x	
(0) identified_key	S	x	
(1) wrapped_key	S	x	
(2) agreed_key	S	x	
key_info_options	O	x	
identified_key_options	S	x	
(0) global_unicast_encryption_key	S	x	
(1) global_broadcast_encryption_key	S	x	
wrapped_key_options	S	x	
kek_id	O	x	
(0) master_key	O	x	
key_ciphered_data	O	x	
agreed_key_options	S	x	
key_parameters	O	x	
key_ciphered_data	O	x	

O: Obligatoire (partie d'une structure).

S: Sélectionnable (partie d'un CHOICE).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 25 – Paramètres de protection de la méthode *get_protected_attributes*

Paramètres de protection	demande	réponse
array protection_parameters_element	O	O (=)
protection_type	O	O (=)
(0) authentification	S	S (=)
(1) cryptage	S	S (=)
(2) authentification et cryptage	S	S (=)
(3) signature numérique	S	S (=)
protection_options	O	O (=)
transaction_id	O	O (=)
originator_system_title	O	O[1]
recipient_system_title	O	O[2]
other_information	O	O (=)
key_info	O	O (=)
key_info_type	O	O (=)
(0) identified_key	S	S (=)
(1) wrapped_key	S	S (=)
(2) agreed_key	S	S (=)
key_info_options	O	O (=)
identified_key_options	S	S (=)
(0) global_unicast_encryption_key	S	S (=)
(1) global_broadcast_encryption_key	S	S (=)
wrapped_key_options	S	S (=)
kek_id	O	O (=)
(0) master_key	O	O (=)
key_ciphered_data	–	O[3]
agreed_key_options	S	S (=)
key_parameters	O	O (=)
key_ciphered_data	–	O[4]

Noter que les paramètres de protection sont issus de la demande et placés dans la réponse. Les paramètres de protection s'appliquent uniquement à la protection de la réponse.

[1] originator_system_title de la demande est placé dans recipient_system_title de la réponse.

[2] recipient_system_title de la demande est placé dans originator_system_title de la réponse.

[3] key_ciphered_data est renseigné par le serveur.

[4] Si le schéma One-pass Diffie-Hellman C(1e, 1s, ECC CDH) est utilisé, cet élément est renseigné par le serveur.

O: Obligatoire (partie d'une structure).

S: Sélectionnable (partie d'un CHOICE).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 26 – Paramètres de protection de la méthode *set_protected_attributes*

Paramètres de protection	demande	réponse
array protection_parameters_element	O	
protection_type	O	
(0) authentification	S	
(1) cryptage	S	
(2) authentification et cryptage	S	
(3) signature numérique	S	
protection_options	O	
transaction_id	O	
originator_system_title	O	
recipient_system_title	O	
other_information	O	
key_info	O	
key_info_type	O	
(0) identified_key	S	
(1) wrapped_key	S	
(2) agreed_key	S	
key_info_options	O	
identified_key_options	S	
(0) global_unicast_encryption_key	S	
(1) global_broadcast_encryption_key	S	
wrapped_key_options	S	
kek_id	O	
(0) master_key	O	
key_ciphered_data	O	
agreed_key_options	S	
key_parameters	O	
key_ciphered_data	O	
Noter que les paramètres de protection sont issus de la demande et sont absents de la réponse. Les paramètres de protection sont uniquement utilisés pour retirer la protection de la demande.		
O: Obligatoire (partie d'une structure). S: Sélectionnable (partie d'un CHOICE).		

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 27 – Paramètres de protection de la méthode *invoke_protected_method*

Paramètres de protection	demande	réponse
array protection_parameters_element	O	O (=)
protection_type	O	O (=)
(0) authentification	S	S(=)
(1) cryptage	S	S(=)
(2) authentification et cryptage	S	S(=)
(3) signature numérique	S	S(=)
protection_options	O	O (=)
transaction_id	O	O (=)
originator_system_title	O	O[1]
recipient_system_title	O	O[2]
other_information	O	O (=)
key_info	O	O (=)
key_info_type	O	O (=)
(0) identified_key	S	S(=)
(1) wrapped_key	S	S(=)
(2) agreed_key	S	S(=)
key_info_options	O	O (=)
identified_key_options	S	S(=)
(0) global_unicast_encryption_key	S	S(=)
(1) global_broadcast_encryption_key	S	S(=)
wrapped_key_options	S	S(=)
kek_id	O	O (=)
(0) master_key	O	O (=)
key_ciphered_data	O	O[3]
agreed_key_options	S	S(=)
key_parameters	O	O (=)
key_ciphered_data	O	O[4]

Noter que les paramètres de protection sont issus de la demande et placés dans la réponse. Les paramètres de protection de la demande sont utilisés pour retirer la protection de la demande, les paramètres de protection de la réponse étant utilisés pour appliquer la protection à la réponse.

[1] originator_system_title de la demande est placé dans recipient_system_title de la réponse.

[2] recipient_system_title de la demande est placé dans originator_system_title de la réponse.

[3] key_ciphered_data est envoyé par l'émetteur dans la demande et renseigné par le serveur dans la réponse.

[4] Si le schéma One-pass Diffie-Hellman C(1e, 1s, ECC CDH) est utilisé, cet élément est renseigné par le serveur.

O: Obligatoire (partie d'une structure).

S: Sélectionnable (partie d'un CHOICE).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.10 Function control (class_id: 122, version: 0)

4.4.10.1 Vue d'ensemble

Les instances de l'IC "Function control" permettent d'activer et de désactiver des fonctions sur le serveur. Chaque fonction qui peut être activée/désactivée est identifiée par un nom et définie par un ensemble particulier d'identifiants d'objet référencés.

Pour activer et désactiver des fonctions commandées par le temps, les objets "Single action schedule" et "Script table" sont également spécifiés.

Function control		**0...n**	**class_id = 122, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. activation_status	(dyn.)	array				x + 0x08
3. function_list	(static)	array				x + 0x10
Méthodes spécifiques		***o/f***				
1. set_function_status (data)		o				x + 0x20
2. add_function (data)		f				x + 0x28
3. remove_function (data)		f				x + 0x30

4.4.10.2 Description d'attribut

4.4.10.2.1 logical_name

Identifie l'instance d'objet "Function control". Voir 6.2.38.

4.4.10.2.2 activation_status

Présente l'état en cours de chaque bloc fonctionnel défini dans l'attribut *function_list*.

```
activation_status::= array function_status_type
function_status_type::= structure
{
function_name    octet-string,
function_status  boolean
}
```

TRUE: fonction activée,

FALSE: fonction désactivée.

4.4.10.3 function_list

Définit une liste de fonctions qui peuvent être activées ou désactivées en appelant la méthode *set_function_status*.

```
function_list::= array    functional_block

functional_block::=       structure
{
function_name:            octet-string,
function_specification:   array function_definition
}

function_definition::= structure
{
class_id:                 long-unsigned,
logical_name:             octet-string
}
```

L'élément function_specification contient une liste d'objets impliqués dans cette fonction. Ils sont référencés par leurs ID de classe et noms logiques.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Dans le cas de fonctions qui ne peuvent pas être liées à des objets particuliers, class_id = 0 est utilisé et le deuxième élément contient un nom descriptif en lieu et place d'un logical_name.

Les noms des fonctions et le comportement du serveur en cas d'activation ou de désactivation d'une fonction doivent être définis dans les spécifications d'accompagnement spécifiques au projet.

Noter également que cet attribut commande uniquement les fonctions disponibles dans le serveur et qui peuvent être activées ou désactivées. Il n'est pas possible de télécharger de nouvelles fonctions dans le serveur en modifiant cet attribut ou en appelant la méthode *add_function*.

4.4.10.4 Description de la méthode

4.4.10.4.1 set_function_status (data)

Permet d'activer ou de désactiver une ou plusieurs fonctions définies dans l'attribut *function_list*.

data::= array function_status_type

tel que défini dans l'attribut *activation_status* (4.4.10.2.2).

L'état des fonctions qui ne sont pas répertoriées n'est pas modifié.

4.4.10.4.2 add_function (data)

Permet d'ajouter une nouvelle fonction à l'attribut *function_list*.

Si une fonction portant le même nom existe déjà, elle est remplacée par la nouvelle fonction.

data::= functional_block

functional_block: voir 4.4.10.3.

4.4.10.4.3 remove_function (data)

Permet de retirer une fonction de l'attribut *function_list*.

data::= octet-string, contenant le function_name.

4.4.11 Array manager (class_id = 123, version = 0)

4.4.11.1 Vue d'ensemble

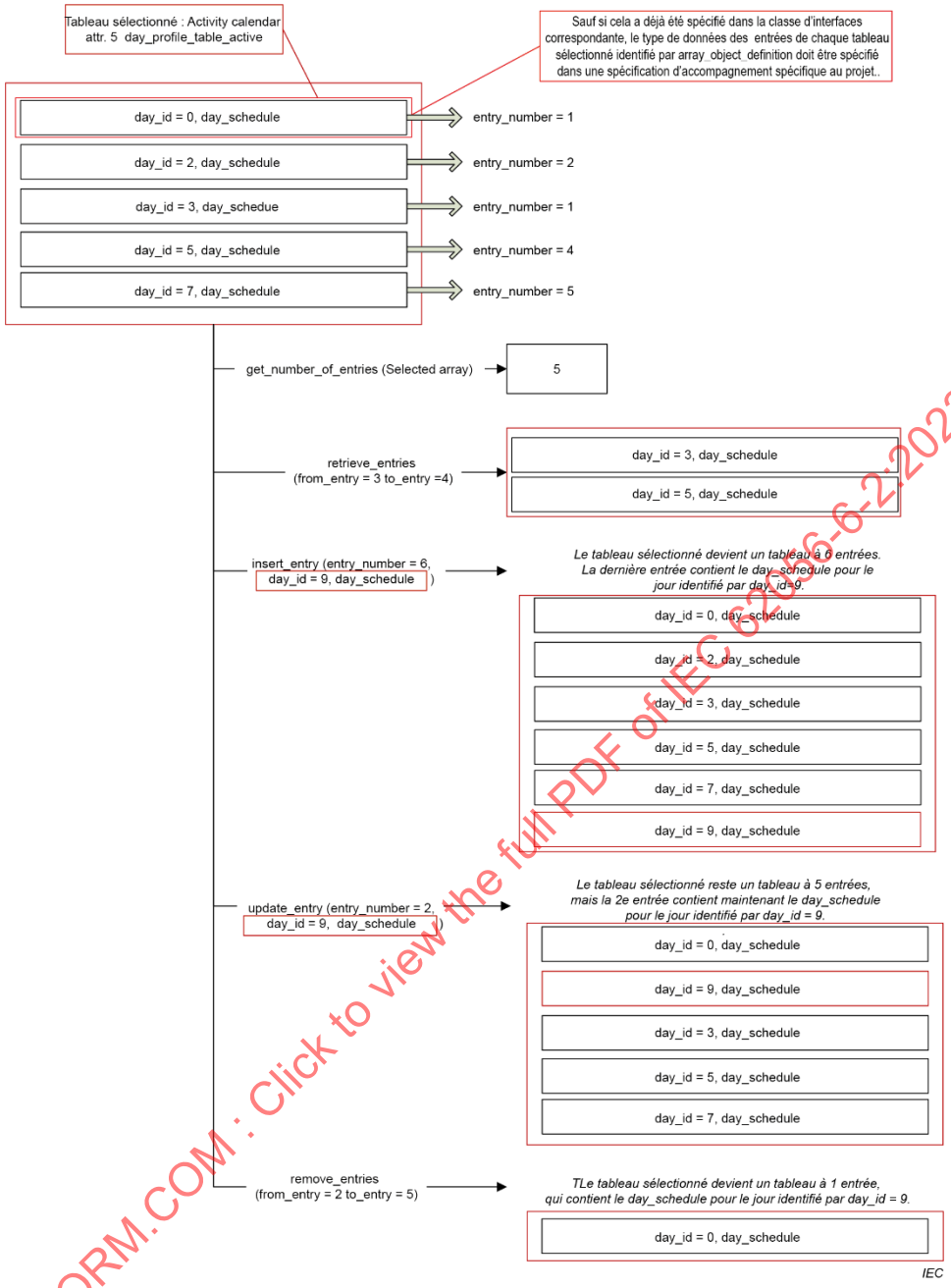
Les instances de l'IC "Array manager" permettent de gérer les attributs de type *array* d'autres objets d'interface, c'est-à-dire:

- de récupérer le nombre d'entrées;
- de lire une plage d'entrées de manière sélective;
- d'insérer une nouvelle entrée ou de mettre à jour une entrée existante;
- de retirer une plage d'entrées.

Chaque instance permet de gérer plusieurs attributs de type *array* qui lui sont attribués.

Un exemple d'application est présenté à la Figure 16.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 16 – Exemple de gestion d'un tableau

Array manager		**0...n**	**class_id = 123, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. array_object_list	(static)	array				x + 0x08
Méthodes spécifiques		***o/f***				
1. retrieve_number_of_entries (data)		o				x + 0x20
2. retrieve_entries (data)		o				x + 0x28
3. insert_entry (data)		f				x + 0x30
4. update_entry (data)		f				x + 0x38
5. remove_entries (data)		f				x + 0x40

4.4.11.2 Description d'attribut

4.4.11.2.1 logical_name

Identifie l'instance d'objet "Array manager". Voir 6.2.16.

4.4.11.2.2 array_object_list

Définit la liste d'attributs de type *array* qui peuvent être gérés par une instance de cette IC. Chaque objet "Array manager" peut gérer de 1 à n tableaux.

```
array        array_object_list_definition

array_object_list_definition::=      structure
{
array_object_id:                     unsigned,
array_object_list_element:           array_object_definition
}
array_object_definition::=     structure
{
class_id:          long-unsigned,
logical_name:      octet_string,
attribute_index:   integer
}
```

où attribute_index est un pointeur vers l'attribut au sein de l'objet identifié par son class_id et son logical_name. Attribute_index 1 fait référence au 1er attribut (c'est-à-dire le logical_name), attribute_index 2 au 2e, etc.

Chaque attribut identifié par array_object_definition doit être de type *array*.

Un attribut de type *array* contient un certain nombre d'entrées qui peuvent être référencées par leur numéro d'entrée. Le numéro 1 identifie la première entrée et le numéro m la dernière entrée, où m est le nombre d'entrées d'un tableau sélectionné.

Sauf si cela a déjà été spécifié dans la classe d'interfaces correspondante, le type de données des entrées de chaque tableau identifié par array_object_definition doit être spécifié dans une spécification d'accompagnement spécifique au projet.

Si le tableau cible est accessible par l'intermédiaire d'un objet "Array manager", ses droits d'accès sont observés.

4.4.11.3 Description de la méthode

4.4.11.3.1 retrieve_number_of_entries (data)

Retourne le nombre d'entrées du tableau identifié.

Le paramètre d'appel de la méthode identifie un tableau de la liste array_object_list par son array_object_id:

```
data::= unsigned
```

Le paramètre de retour contient le nombre d'entrées du tableau identifié.

4.4.11.3.2 retrieve_entries (data)

Retourne une plage d'entrées dans l'un des tableaux de array_object_list.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Les paramètres d'appel de la méthode identifient un tableau de array_object_list par son array_object_id et les entrées à récupérer:

```
data::= entries_to_retrieve

entries_to_retrieve::=   structure
{
array_object_id:  unsigned,
list_of_entries:    structure
{
        from_entry:         long-unsigned,
        to_entry:           long-unsigned
}
}
```

Où:

– array_object_id identifie l'un des tableaux gérés par un objet "Array manager";

– list_of_entries identifie les entrées à récupérer.

Si le nombre d'entrées du tableau sélectionné est m:

– si from_entry < last_entry < m, les entrées de cette plage sont récupérées;

– si from_entry < last_entry, mais que last_entry > m, les entrées identifiées par from_entry jusqu'à last_entry sont récupérées;

– si from_entry < last_entry, mais que from_entry > m, la méthode retourne un tableau de 0 élément;

– si from_entry > last_entry, l'appel de la méthode n'aboutit pas.

Les paramètres de retour contiennent un tableau des entrées sélectionnées:

```
data::=       retrieved_entries:

retrieved_entries:      array entry

entry::= attribute specific
```

Le type de données dépend de l'attribut de type *array* indiqué dans la spécification de l'IC ou dans la spécification d'accompagnement spécifique au projet.

4.4.11.3.3 insert_entry (data)

Permet d'insérer une nouvelle entrée dans un tableau sélectionné.

```
data::=             structure
{
array_object_id:        unsigned,
entry_to_insert:        structure
        {
                entry_number:   long-unsigned,
                entry:          attribute specific

        Le type de données de l'entrée dépend de l'attribut de type array indiqué dans
        la spécification de l'IC ou dans la spécification d'accompagnement spécifique
        au projet.
        }
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Où:

– array_object_id identifie l'un des tableaux gérés par un objet "Array manager";
– entry_to_insert identifie l'entry_number et contient l'entrée elle-même à insérer.

Si entry_number = 0:

– si le tableau sélectionné est déjà complet, l'appel de la méthode n'aboutit pas;
– sinon, la nouvelle entrée est insérée au début du tableau: elle devient la première entrée et toutes les autres entrées du tableau sélectionné sont décalées de un vers le haut;

Si entry_number > m (où m est le nombre d'entrées du tableau):

– si le tableau sélectionné est déjà complet, l'appel de la méthode n'aboutit pas;
– sinon, la nouvelle entrée est insérée à la fin du tableau sélectionné: elle devient la dernière entrée. Les autres entrées ne sont pas décalées.

Si un entry_number identifie une entrée existante, il est inséré après l'entrée existante.

Si les entrées sont décalées à la suite de l'insertion d'une nouvelle entrée, toutes les références les concernant doivent être mises à jour.

4.4.11.3.4 update_entry (data)

Permet de mettre à jour une entrée existante dans un tableau sélectionné.

```
data::=              structure
{
array_object_id:         unsigned,
entry_to_update:         structure
      {
      entry_number:      long-unsigned,
      entry:             attribute specific
```

Le type de données de l'entrée dépend de l'attribut de type *array* indiqué dans la spécification de l'IC ou dans la spécification d'accompagnement spécifique au projet.

```
      }
}
```

Où:

– array_object_id identifie l'un des tableaux gérés par un objet "Array manager";
– entry_to_update identifie l'entry_number et contient l'entrée elle-même à mettre à jour.

Si un entry_number identifie une entrée existante, il est écrasé.

Dans tous les autres cas, l'appel de la méthode n'aboutit pas.

4.4.11.3.5 remove_entries (data)

Permet de retirer une plage d'entrées de l'un des tableaux de array_object_list.

Les paramètres d'appel de la méthode identifient un tableau dans array_object_list par son array_object_id et les entrées à retirer:

```
data::= entries_to_remove

entries_to_remove::=    structure
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
{
array_object_id:  unsigned;
list_of_entries:  structure
{
      from_entry:       long-unsigned,
      to_entry:         long-unsigned
}
```

Où:

– array_object_id identifie l'un des tableaux gérés par un objet "Array manager";

– list_of_entries identifie les entrées à retirer.

Si le nombre d'entrées du tableau sélectionné est m:

– si from_entry < last_entry < m, les entrées de cette plage sont retirées;

– si from_entry < last_entry, mais que last_entry > m, les entrées identifiées par from_entry jusqu'à last_entry sont retirées;

– si from_entry < last_entry, mais que from_entry > m, aucune entrée n'est retirée;

– si from_entry > last_entry, l'appel de la méthode n'aboutit pas.

Si les entrées sont décalées à la suite de la suppression d'entrées, toutes les références les concernant doivent être mises à jour.

4.4.12 Communication port protection (class_id = 124, version = 0)

4.4.12.1 Vue d'ensemble

Les instances de l'IC "Communication port protection" peuvent être utilisées pour protéger les ports de communication des serveurs DLMS®/COSEM contre les tentatives de communication potentiellement malveillantes, en particulier pour empêcher les attaques par force brute en réduisant le nombre possible de tentatives.

Chaque instance fait référence à un seul port de communication par son nom logique (code d'OBIS). Si un nombre acceptable de tentatives non réussies est dépassé, le port de communication est alors temporairement bloqué. Le temps de verrouillage peut augmenter à chaque tentative non réussie, jusqu'à ce qu'un temps de verrouillage maximal soit atteint.

Une tentative non réussie est celle qui conduit à rejeter l'APDU transportant une demande de service. Les critères de détection d'une tentative non réussie ne relèvent pas du domaine d'application du présent document.

Les objets comptent à la fois le nombre de tentatives non réussies entre deux réinitialisations et le nombre cumulé de tentatives non réussies.

Il est possible de configurer les ports de communication de telle sorte qu'ils soient bloqués verrouillés à toutes les tentatives ou débloqués à toutes les tentatives.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Communication port protection		0...n	class_id = 124, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	***Nom court***
1. logical_name	(static)	octet-string				x
2. protection_mode	(static)	enum			1	x + 0x08
3. allowed_failed_attempts	(static)	long-unsigned				x + 0x10
4. initial_lockout_time	(static)	double-long-unsigned				x + 0x18
5. steepness_factor	(static)	unsigned			1	x + 0x20
6. max_lockout_time	(static)	double-long-unsigned				x + 0x28
7. port_reference	(static)	octet-string				x + 0x30
8. protection_status	(dyn.)	enum				x + 0x38
9. failed_attempts	(dyn.)	double-long-unsigned				x + 0x40
10. cumulative_failed_attempts	(dyn.)	double-long-unsigned				x + 0x48
Méthodes spécifiques	***o/f***					
1. reset (data)	f	f				x + 0x50

4.4.12.2 Description d'attribut

4.4.12.2.1 logical_name

Identifie l'instance de l'objet "Communication port protection". Voir 6.2.39.

4.4.12.2.2 protection_mode

Contrôle le mode de protection.

enum

(0) locked,
(1) locked_on_failed_attempts,
(2) unlocked

Lorsqu'il est mis sur "locked", aucune communication ne peut avoir lieu via le port référencé par l'attribut 7.

Lorsqu'il est mis sur "locked_on_failed_attempts", le port devient temporairement bloqué lorsque le nombre de tentatives de communication non réussies dépasse un nombre autorisé. La durée de la période de verrouillage peut augmenter à chaque tentative non réussie, comme déterminé par les attributs 5 et 6.

Lorsqu'il est mis sur "unlocked", le mécanisme de verrouillage est désactivé; la communication par l'intermédiaire du port référencé par l'attribut 7 est toujours possible.

Lorsque le port est bloqué, aucune communication ne peut avoir lieu, même les APDU valides sont rejetées.

4.4.12.2.3 allowed_failed_attempts

Contient le nombre de tentatives de communication non réussies autorisées avant le déclenchement du mécanisme de verrouillage.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.12.2.4 initial_lockout_time

Contient la valeur initiale du temps de verrouillage, en secondes, après la première tentative de communication non réussie lorsque la valeur de allowed_failed_attempts est atteinte.

4.4.12.2.5 steepness_factor

Contient un facteur qui contrôle la façon dont le temps de verrouillage est augmenté à chaque tentative non réussie lorsque allowed_failed_attempts est atteint, jusqu'à ce que max_lockout_time soit atteint.

Le temps de verrouillage actuel est calculé en utilisant les formules suivantes:

$$NCA = failed_attempts - allowed_failed_attempts$$

$$CLT = initial_lockout_time \times (steepness_factor^{(NCA-1)})$$

Où:

– NCA est le nombre de tentatives non réussies comptées, c'est-à-dire le nombre de tentatives prises en compte pour allonger le temps de verrouillage;

– CLT est le temps de verrouillage actuel.

Le Tableau 28 représente des exemples de valeurs avec un temps de verrouillage initial de 60 s.

Tableau 28 – Exemples de valeurs pour NCA et CLT

NCA	steepness_factor		
	1	2	3
	CLT (en secondes)		
1	60	60	60
2	60	120	180
3	60	240	540
4	60	480	1 620
5	60	960	4 860

Une fois le verrouillage effectué, même des tentatives valides sont rejetées jusqu'à expiration du temps de verrouillage actuel.

4.4.12.2.6 max_lockout_time

Contient la durée maximale, en secondes, pendant laquelle le port de communication peut être bloqué, même si le nombre de tentatives non réussies ne cesse d'augmenter. Le but de cet attribut est d'éviter une attaque par déni de service.

4.4.12.2.7 port_reference

Contient le nom logique d'un objet qui identifie le port de communication protégé. Cet objet peut être un objet de configuration de port de communication ou tout autre objet choisi de manière appropriée.

Si cette information n'est pas disponible ou nécessaire, l'attribut peut être laissé vide (octet-string [0]).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.4.12.2.8 protection_status

Contient l'état de protection actuel du port de communication.

enum

(0) unlocked,
(1) temporarily_locked,
(2) locked.

L'état est "unlocked" si protection_mode est "unlocked" ou si le nombre de tentatives non réussies ne dépasse pas le nombre de allowed_failed_attempts.

NOTE 1 L'état devient de nouveau "unlocked" lorsque le current_lockout_time s'est écoulé.

L'état est "temporarily_locked" si le nombre de tentatives de communication non réussies dépasse allowed_failed_attempts et si current_lockout_time n'a pas encore expiré.

L'état est "locked" si le protection_mode est "locked".

NOTE 2 Lorsqu'un port est bloqué, le serveur reste accessible sur un autre port.

failed_attempts Contient le nombre total de tentatives non réussies depuis la dernière réinitialisation, indépendamment du protection_mode et du déclenchement du mécanisme de protection.

4.4.12.2.9 cumulative_failed_attempts

Contient le nombre cumulé de tentatives non réussies, indépendamment du protection_mode et du déclenchement du mécanisme de protection.

Cet attribut n'est jamais réinitialisé.

4.4.12.3 Description de la méthode

4.4.12.3.1 reset (data)

Réinitialise les variables du mécanisme de verrouillage:

– la valeur de failed_attempts est réinitialisée à 0;
– le temps de verrouillage actuel est réinitialisé à 0;
– protection_status est positionné à "(0) unlocked" si le protection_mode est (1) lock_on_failed_attempts, et n'est pas affecté pour le reste.

data::= integer (0)

4.5 Classes d'interfaces pour commande à limite temporelle et événementielle

4.5.1 Clock (class_id = 8, version = 0)

4.5.1.1 Vue d'ensemble

Cette IC modélise l'horloge, en gérant toutes les informations relatives à la date et à l'heure, y compris les décalages de l'heure locale par rapport à une référence temporelle généralisée (TUC), en raison des fuseaux horaires et des schémas de changement d'horaire légaux. L'IC offre également diverses méthodes pour régler l'horloge.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Les informations *date* comprennent les éléments year (année), month (mois), day of month (jour dans le mois) et day of week (jour dans la semaine). Les informations *time* comprennent les éléments hour (heure), minutes, seconds (secondes), hundredths of seconds (centièmes de seconde) et le décalage de l'heure locale par rapport au temps TUC. La fonction heure (été/hiver) de changement d'horaire légal modifie le décalage de l'heure locale par rapport au TUC en fonction des attributs (voir Figure 17).

Le point initial et le point final de la fonction en question sont normalement fixés une seule fois. Un algorithme interne calcule le point réel de commutation en fonction de ces valeurs fixées.

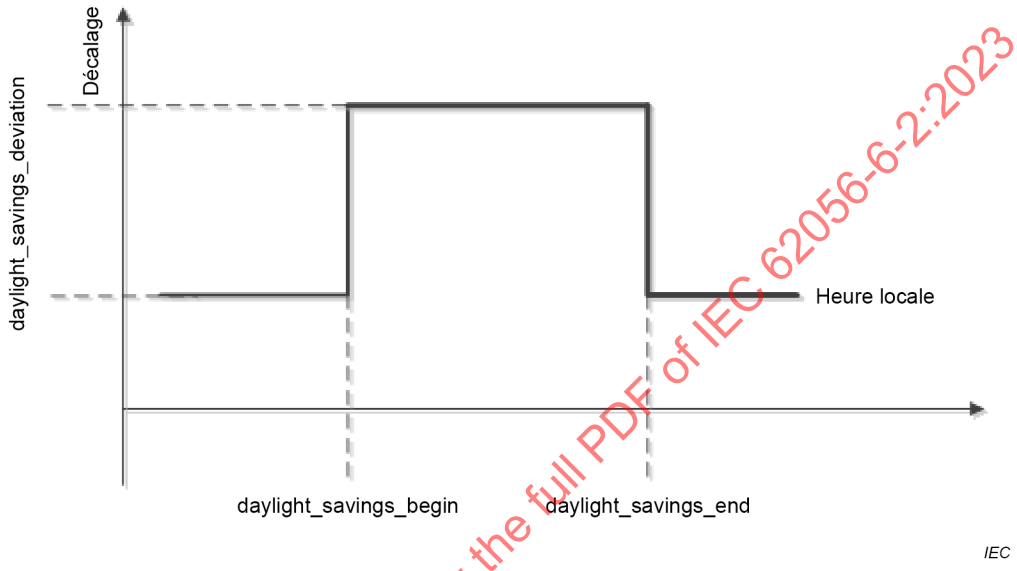


Figure 17 – Concept de temps généralisé

Clock		**0...n**	**class_id = 8, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. time	(dyn.)	octet-string				x + 0x08
3. time_zone	(static)	long	-720	+840		x + 0x10
4. status	(dyn.)	unsigned				x + 0x18
5. daylight_savings_begin	(static)	octet-string				x + 0x20
6. daylight_savings_end	(static)	octet-string				x + 0x28
7. daylight_savings_deviation	(static)	integer	-120	+120		x + 0x30
8. daylight_savings_enabled	(static)	boolean				x + 0x38
9. clock_base	(static)	enum				x + 0x40
Méthodes spécifiques		***o/f***				
1. adjust_to_quarter (data)		f				x + 0x60
2. adjust_to_measuring_period (data)		f				x + 0x68
3. adjust_to_minute (data)		f				x + 0x70
4. adjust_to_preset_time (data)		f				x + 0x78
5. preset_adjusting_time (data)		f				x + 0x80
6. shift_time (data)		f				x + 0x88

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.1.2 Description d'attribut

4.5.1.2.1 logical_name

Identifie l'instance de l'objet "Clock". Voir 6.2.5.

4.5.1.2.2 time

Contient la date et l'heure locales du compteur, son écart par rapport au TUC et l'état.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

Si cet attribut est défini, tous les champs de la chaîne d'octets représentant *date-time* doivent être évalués, et la date et l'heure locales du compteur doivent être définies selon les règles définies en 4.1.6.1.

Seuls les champs spécifiés de *date-time* sont modifiés.

EXEMPLE Pour régler la date sans modifier l'heure, tous les octets relatifs à l'heure dans la chaîne d'octets représentant *date-time* doivent être mis à "not specified" (non spécifié).

4.5.1.2.3 time_zone

L'écart de l'heure locale normale par rapport au TUC en minutes. La valeur dépend de la position géographique du compteur.

4.5.1.2.4 status

clock_status géré par le compteur. L'élément clock_status indiqué dans l'attribut *time* est égal à la valeur de cet attribut.

unsigned, mis en forme tel que spécifié en 4.1.6.1 pour *clock_status*

4.5.1.2.5 daylight_savings_ begin

Définit la date et l'heure locales de basculement lorsque l'heure locale commence à être décalée par rapport à l'heure normale.

Pour des définitions génériques, il est permis d'utiliser des caractères génériques.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

4.5.1.2.6 daylight_savings_end

Définit la date et l'heure locales de basculement lorsque l'heure locale finit d'être décalée par rapport à l'heure normale.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

4.5.1.2.7 daylight_savings_ deviation

Contient le nombre de minutes dont l'écart d'heure généralisée doit être corrigé au daylight_savings_begin (moment du début d'heure [été/hiver]).

integer: Écart dans la plage de ± 120 min

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.1.2.8 daylight_savings_ enabled

boolean: TRUE: DST enabled (heure hiver/été activée),

FALSE: DST disabled (heure hiver/été désactivée).

NOTE Ce paramètre permet d'activer et de désactiver la fonction d'heure d'été/hiver. L'état en cours est indiqué dans l'attribut *status*.

4.5.1.2.9 clock_base

Définit la provenance des informations temporelles de base.

enum:

(0) non définie,
(1) quartz interne,
(2) fréquence secteur de 50 Hz,
(3) fréquence secteur de 60 Hz,
(4) GPS (système de positionnement global),
(5) radiocommandée.

4.5.1.3 Description de la méthode

4.5.1.3.1 adjust_to_quarter (data)

Règle l'heure du compteur sur le plus proche (+/-) quart d'heure (*:00, *:15, *:30, *:45).

data::= integer (0)

4.5.1.3.2 adjust_to_measuring_period (data)

Règle l'heure du compteur sur le plus proche (+/-) point de départ d'une période de mesure.

data::= integer (0)

4.5.1.3.3 adjust_to_minute (data)

Règle l'heure du compteur sur la minute la plus proche.

Si second_counter < 30 s, second_counter est mis à 0.

Si second_counter ≥ 30 s, second_counter est mis à 0 et les valeurs de minute_counter et toutes les valeurs d'horloge dépendantes sont incrémentées si nécessaire.

data::= integer (0)

4.5.1.3.4 adjust_to_preset_ time (data)

Cette méthode est utilisée conjointement avec la méthode preset_adjusting_time. Si l'heure du compteur s'inscrit entre validity_interval_start et validity_interval_end, l'heure est mise à preset_time.

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.1.3.5 preset_adjusting_ time (data)

Prérègle l'heure sur une nouvelle valeur (preset_time) et définit un validity_interval dans lequel la nouvelle heure peut être activée.

```
data::= structure
{
preset_time:                octet-string,
validity_interval_start:    octet-string,
validity_interval_end:      octet-string
}
```

tous les octet-string, formatés comme indiqué en 4.6.1 pour *date-time*.

4.5.1.3.6 shift_time (data)

Décale l'heure de *n* (-900 <= *n* <= 900) s.

```
data::= long
```

4.5.2 Script table (class_id = 9, version = 0)

4.5.2.1 Vue d'ensemble

Cette IC permet de modéliser le déclenchement d'une série d'actions en exécutant des scripts par la méthode *execute* (data).

Les objets "Script table" contiennent un tableau d'entrées de scripts. Chaque entrée consiste en un identifiant de script ("script identifier") et une série de spécifications d'actions ("action specifications"). Une spécification d'action active une méthode ou modifie un attribut d'un objet COSEM au sein d'un dispositif logique.

Un certain script peut être activé par d'autres objets COSEM au sein du même dispositif logique ou depuis l'extérieur.

Si deux scripts sont à exécuter dans la même instance temporelle, celui ayant l'indice le plus petit est exécuté en premier.

Script table	**0...n**	**class_id = 9, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. scripts (static)	array				x + 0x08
Méthodes spécifiques	***o/f***				
1. execute (data)	o				x + 0x20

4.5.2.2 Description d'attribut

4.5.2.2.1 logical_name

Identifie l'instance de l'objet "Script table". Voir 6.2.7.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.2.2.2 scripts

Spécifie les différents scripts, c'est-à-dire la liste d'actions.

```
array        script

script::= structure
{
script_identifier:  long-unsigned,
actions:            array action_specification
}
```

Le script_identifier 0 est réservé. S'il est spécifié avec une méthode *execute*, il donne un script vide (aucune action à accomplir).

```
action_specification::= structure
{
      service_id:           enum,
      class_id:             long-unsigned,
      logical_name:         octet-string,
      index:                integer,
      parameter:            spécifique au service
}
```

Où:

– l'élément service_id définit l'action à appliquer à l'objet référencé:

  (1) écrire un attribut;

  (2) exécuter une méthode spécifique;

– l'élément index définit (avec service_id 1) quel attribut de l'objet sélectionné est affecté ou (avec service_id 2) quelle méthode spécifique est à exécuter. Le premier attribut (logical_name) a l'indice 1, la première méthode spécifique a l'indice 1 également.

NOTE 1 L'action_specification se limite à activer des méthodes qui ne produisent pas de réponse (du serveur au client).

NOTE 2 Une spécification d'action "dummy" (fictive) avec tous les éléments ayant la valeur 0 signifie que l'action n'est pas configurée.

4.5.2.3 Description de la méthode

4.5.2.3.1 execute (data)

Exécute le script spécifié dans les données de paramètre.

```
data::= long-unsigned
```

Si *data* concorde avec l'un des script_identifiers dans le tableau de scripts, l'action_specification correspondante est exécutée.

4.5.3 Schedule (class_id = 10, version = 0)

4.5.3.1 Vue d'ensemble

Cette IC, associée à l'IC "Special days table", permet de modéliser des activités qui dépendent de l'heure et de la date au sein d'un dispositif. Le Tableau 29 et le Tableau 30 donnent une vue d'ensemble et présentent les interactions entre les deux IC.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 29 – Schedule

Indice	enable	action (script)	Switch _time	validity _windo w	exec_weekdays							exec_specdays					date range	
					Mo	Tu	We	Th	Fr	Sa	Su	S1	S2	…	S8	S9	begin_ date	end_ date
120	Oui	xxxx:yy	06:00	0xFFFF	*x*	*x*	*x*	*x*	*x*	*x*							xx-04-01	xx-09-30
121	Oui	xxxx:yy	22:00	15	*x*	*x*	*x*	*x*	*x*								xx-04-01	xx-09-30
122	Oui	xxxx:yy	12:00	0						*x*							xx-04-01	xx-09-30
200	Non	xxxx:yy	06:30		*x*	*x*	*x*	*x*	*x*	*x*							xx-04-01	xx-09-30
201	Non	xxxx:yy	21:30		*x*	*x*	*x*	*x*	*x*								xx-04-01	xx-09-30
202	Non	xxxx:yy	11:00							*x*							xx-04-01	xx-09-30

Tableau 30 – Special days table

Indice	special_day_date	day_id
12	xx-12-24	S1
33	xx-12-25	S3
77	97-03-31	S3

Schedule		**0…n**	**class_id = 10, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. entries	(static)	array				x + 0x08
Méthodes spécifiques		***o/f***				
1. enable/disable (data)		f				x + 0x20
2. insert (data)		f				x + 0x28
3. delete (data)		f				x + 0x30

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.3.2 Description d'attribut

4.5.3.2.1 logical_name

Identifie l'instance de l'objet "Schedule". Voir 6.2.9.

4.5.3.2.2 entries

Spécifie les scripts devant être exécutés à des instants donnés. Un seul script peut être exécuté par entrée.

```
array        schedule_table_entry
schedule_table_entry::= structure
{
index:                  long-unsigned,
enable:                 boolean,
script_logical_name:    octet-string,
script_selector:        long-unsigned,
switch_time:            octet-string,
validity_window:        long-unsigned,
exec_weekdays:          bit-string,
exec_specdays:          bit-string,
```

```
begin_date:          octet-string,
end_date:            octet-string
}
```

Où:

– script_logical_name: définit le nom logique de l'objet "Script table";

– script_selector: définit le script_identifier du script à exécuter;

– switch_time accepte les caractères génériques pour définir les entrées répétitives. Le format de l'octet-string suit les règles établies en 4.1.6.1 pour *time*;

– validity_window définit une période, en minutes, pendant laquelle une entrée doit être traitée après une coupure secteur. (temps entre le switch_time défini et le power_up effectif) 0xFFFF: le script est traité à tout instant;

– exec_weekdays définit les jours de la semaine pendant lesquels l'entrée est valide;

– exec_specdays établit le lien avec l'IC "Special days table", day_id;

begin_date et end_date définissent la date à laquelle l'entrée est valide (les caractères génériques sont permis). Le format suit les règles établies en 4.1.6.1 pour *date*.

4.5.3.3 Description de la méthode

4.5.3.3.1 enable/disable (data)

Met le bit désactivé des entrées de la plage A à *true* puis active les entrées de la plage B.

```
data::= structure
{
firstIndexA:      long-unsigned,
lastIndexA:       long-unsigned,
firstIndexB:      long-unsigned,
lastIndexB:       long-unsigned
}
```

Où:

– firstIndexA premier indice de la plage qui est désactivée;

– lastIndexA dernier indice de la plage qui est désactivée;

– firstIndexB premier indice de la plage qui est activée;

– lastIndexB dernier indice de la plage qui est activée;

– firstIndexA/B < lastIndexA/B: toutes les entrées de la plage A/B sont désactivées/activées;

– firstIndexA/B = lastIndexA/B: une entrée est désactivée/activée;

– firstIndexA/B > lastIndexA/B: rien n'est désactivé/activé;

– firstIndexA/B et lastIndexA/B > 9999: aucune entrée n'est désactivée/activée.

4.5.3.3.2 insert (data)

Insère une nouvelle entrée dans le tableau. Si l'indice de l'entrée existe déjà, l'entrée existante est écrasée par la nouvelle entrée.

```
entry: schedule_table_entry

data::=      correspondant à l'entrée
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.3.3.3 delete (data)

Efface une plage d'entrées dans le tableau.

```
data::= structure
{
firstIndex:   long-unsigned,
lastIndex:    long-unsigned
}
```

Où:

– firstIndex: premier indice de la plage qui est effacée;

– lastIndex: dernier indice de la plage qui est effacée;

– firstIndex < lastIndex: toutes les entrées de la plage A/B sont effacées;

– firstIndex = lastIndex: une entrée est effacée;

– firstIndex > lastIndex: rien n'est effacé.

Remarques concernant les "incohérences" dans les entrées du tableau:

S'il convient d'exécuter plusieurs fois le même script à une instance de *time* spécifique, il n'est exécuté qu'une seule fois.

S'il convient d'exécuter des scripts différents dans la même instance de *time*, l'ordre d'exécution est selon l'indice. Le script à l'indice le plus bas est exécuté en premier.

4.5.3.4 Reprise après coupure secteur

Après une coupure secteur, le programme tout entier est traité pour exécuter tous les scripts nécessaires qui seraient perdus pendant une coupure secteur. À cet effet, les entrées qui n'ont pas été exécutées pendant la coupure secteur doivent être détectées. Selon la fenêtre de validité, elles sont exécutées dans l'ordre correct (comme elles l'auraient été en fonctionnement normal).

4.5.3.5 Gestion des changements de temps

Il existe quatre différentes "actions" de changements de temps:

a) réglage de l'heure en avance (en écrivant dans l'attribut *time* de l'objet "Clock");

b) réglage de l'heure en retard (en écrivant dans l'attribut *time* de l'objet "Clock");

c) synchronisation du temps (en utilisant la méthode *adjust_to_quarter* de l'objet "Clock");

d) économie d'énergie (heure d'été/hiver).

Ces quatre actions nécessitent une gestion différente exécutée par le programme en interaction avec l'activité de réglage de l'heure.

4.5.3.6 Réglage de l'heure en avance

Il est géré de la même façon qu'une coupure secteur. Toutes les entrées manquées sont exécutées en fonction de validity_window. Un réglage de temps court (défini et spécifique au fabricant) peut être géré comme une synchronisation du temps.

4.5.3.7 Réglage de l'heure en retard

Il se traduit par une répétition des entrées qui sont activées pendant le temps répété. Un réglage de temps court (défini et spécifique au fabricant) peut être géré comme une synchronisation du temps.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.3.8 Synchronisation du temps

La synchronisation du temps est utilisée pour corriger de petits écarts entre l'horloge maîtresse et l'horloge locale. L'algorithme est spécifique au fabricant. Il doit garantir qu'aucune entrée du programme ne se perd ou n'est exécutée deux fois. L'attribut validity_window n'a aucun effet, car toutes les entrées sont à exécuter en fonctionnement normal.

4.5.3.9 Heure (été/hiver) de changement d'horaire légal

Si l'horloge est avancée, tous les scripts qui s'inscrivent dans l'intervalle d'avance (et seraient donc perdus) sont exécutés. Si l'horloge est reculée, la réexécution des scripts qui s'inscrivent dans l'intervalle de retard est arrêtée.

4.5.4 Special days table (class_id = 11, version = 0)

4.5.4.1 Vue d'ensemble

Cette IC permet de définir des dates spéciales. A ces dates, un comportement de commutation spécial prévaut sur le comportement normal. L'IC fonctionne conjointement avec la classe "Schedule" ou "Activity calendar". L'élément de donnée de liaison est *day_id*.

Special days table	**0...n**	**class_id = 11, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. entries (static)	array				x + 0x08
Méthodes spécifiques	***o/f***				
1. insert (data)	f				x + 0x10
2. delete (data)	f				x + 0x18

4.5.4.2 Description d'attribut

4.5.4.2.1 logical_name

Identifie l'instance de l'objet "Special days table". Voir 6.2.8.

4.5.4.2.2 entries

Spécifie un identifiant de jour spécial pour une date donnée. La date peut comporter des caractères génériques pour les jours spéciaux qui se répètent, comme Noël.

```
array        spec_day_entry
spec_day_entry::= structure
{
index:                  long-unsigned,
specialday_date:        octet-string,
day_id:                 unsigned
}
```

Où:

– le formatage de specialday_date suit les règles établies en 4.6.1 pour *date*;

– la plage du day_id doit concorder avec la longueur de la chaîne binaire exec_specdays dans l'objet associé de l'IC "Schedule".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.4.3 Description de la méthode

4.5.4.3.1 insert (data)

Insère une nouvelle entrée dans le tableau.

entry::= spec_day_entry

Si un jour spécial ayant le même indice ou la même date qu'un jour déjà défini est inséré, l'ancienne entrée sera écrasée.

4.5.4.3.2 delete (data)

Efface une entrée dans le tableau.

index Indice de l'entrée qui doit être supprimée.

data::= long-unsigned

4.5.5 Activity calendar (class_id = 20, version = 0)

4.5.5.1 Vue d'ensemble

Cette IC permet de modéliser la gestion de diverses structures de tarification dans le compteur. L'IC fournit une liste d'actions programmées, suivant le format classique des programmes basés sur un calendrier en définissant des saisons, des semaines, etc.

Un objet "Activity calendar" peut coexister avec l'objet "Schedule" plus général et peut même le chevaucher. Si les actions d'un objet "Schedule" sont programmées pour la même heure d'activation que dans un objet "Activity calendar", les actions déclenchées par l'objet "Schedule" sont exécutées en premier.

Après une coupure secteur, la dernière action ("last action") manquée issue de l'objet "Activity calendar" est exécutée (en différé). Le but est d'assurer une tarification correcte après une mise sous tension. Si un objet "Schedule" est présent, la "dernière action" manquée de l'objet "Activity calendar" doit être exécutée à l'heure correcte dans la séquence d'actions demandées par l'objet "Schedule".

L'objet "Activity calendar" définit l'activation de certains scripts, qui peuvent accomplir différentes activités à l'intérieur du dispositif logique. L'interface avec l'IC "Script table" est la même que dans le cas de l'IC "Schedule" (voir 4.5.3).

Si une instance de l'IC "Special days table" (voir 4.5.4) est disponible, les entrées pertinentes prévalent sur la sélection d'un profil de jour commandée par l'objet "Activity calendar". Le profil de jour référencé dans le "Special days table" active le day_schedule du *day_profile_table* dans l'objet "Activity calendar" par référencement par le biais du day_id.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Activity calendar		0...n	class_id = 20, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. calendar_name_active	(static)	octet-string				x + 0x08
3. season_profile_active	(static)	array				x + 0x10
4. week_profile_table_active	(static)	array				x + 0x18
5. day_profile_table_active	(static)	array				x + 0x20
6. calendar_name_passive	(static)	octet-string				x + 0x28
7. season_profile_passive	(static)	array				x + 0x30
8. week_profile_table_passive	(static)	array				x + 0x38
9. day_profile_table_passive	(static)	array				x + 0x40
10. activate_passive_calendar_time	(static)	octet-string				x + 0x48
Méthodes spécifiques		***o/f***				
1. activate_passive_calendar (data)		f				x + 0x50

4.5.5.2 Description d'attribut

NOTE Les attributs appelés ..._active sont actuellement actifs. Ceux appelés ..._passive seront activés par la méthode spécifique activate_passive_calendar.

4.5.5.2.1 logical_name

Identifie l'instance de l'objet "Activity calendar". Voir 6.2.10.

4.5.5.2.2 calendar_name_active

Contient généralement un identifiant qui est descriptif du jeu de scripts activés par l'objet.

4.5.5.2.3 season_profile_active

Contient une liste de saisons définies par leur date de début et un week_profile spécifique devant être exécuté. La liste est triée selon season_start (par ordre croissant);

```
array        season

season::= structure
{
season_profile_name:        octet-string,
season_start:               octet-string,
week_name:                  octet-string
}
```

Où:

– season_profile_name est un nom défini par l'utilisateur qui identifie le season_profile courant;

– season_start définit l'heure de début de la saison, mise en forme comme spécifié en 4.6.1 pour *date-time*. Les caractères génériques sont autorisés dans season_start. Si tous les champs sont des caractères génériques, la saison ne commencera jamais. L'utilisation de caractères génériques exige un soin particulier pour éviter tout paramétrage conflictuel, c'est-à-dire éviter que deux saisons différentes aient le même season_start;

NOTE La saison en cours est terminée par le season_start de la saison suivante.

– week_name définit le week_profile actif dans cette saison.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.5.2.4 week_profile_table_active

Contient un tableau de week_profiles à utiliser dans les différentes saisons. Pour chaque week_profile, le day_profile pour chaque jour d'une semaine est identifié.

```
array        week_profile

week_profile::= structure
{
week_profile_name:    octet-string,
monday:               day_id,
tuesday:              day_id,
wednesday:            day_id,
thursday:             day_id,
friday:               day_id,
saturday:             day_id,
sunday:               day_id
}

day_id: unsigned
```

Où:

– week_profile_name est un nom défini par l'utilisateur qui identifie le week_profile courant;

– Monday définit le day_profile valide le lundi;

– …

Sunday définit le day_profile valide le dimanche.

4.5.5.2.5 day_profile_table_active

Contient un tableau de day_profiles, identifiés par leur day_id. Pour chaque day_profile, une liste d'actions programmées est définie par un script à exécuter et l'heure d'activation (start_time) correspondante. La liste est triée selon start_time.

```
array        day_profile

day_profile::= structure
{
day_id:              unsigned,
day_schedule:        array day_profile_action
}

day_profile_action::= structure
{
start_time:          octet-string,
script_logical_name: octet-string,
script_selector:     long-unsigned
}
```

Où:

– day_id est un identifiant défini par l'utilisateur, qui identifie le day_profile courant;

– start_time: définit l'heure à laquelle le script est à exécuter (aucun caractère générique). Le format suit les règles établies en 4.1.6.1 pour *time*;

– script_logical_name: définit le *logical_name* de l'objet "Script table";

– script_selector: définit le script_identifier du script à exécuter.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.5.2.6 activate_passive_calendar_time

Définit l'heure à laquelle l'objet lui-même appelle la méthode spécifique *activate_passive_calendar*. Une définition avec une notation "not specified" (non spécifié) dans tous les champs de l'attribut désactivera cet automatisme. Une notation "not specified" partielle, c'est-à-dire juste dans quelques champs de la date et de l'heure, n'est pas autorisée.

octet-string, mis en forme comme indiqué en 4.6.1 pour *date-time*.

4.5.5.2.7 calendar_name_passive

Contient généralement un identifiant qui est descriptif du jeu de scripts activés par l'objet.

4.5.5.2.8 season_profile_passive

Contient une liste de saisons définies par leur date de début et un week_profile spécifique devant être exécuté. La liste est triée selon season_start (par ordre croissant).

```
array        season

season::= structure
{
season_profile_name:        octet-string,
season_start:               octet-string,
week_name:                  octet-string
}
```

Où:

– season_profile_name est un nom défini par l'utilisateur qui identifie le season_profile courant;

– season_start définit l'heure de début de la saison, mise en forme comme spécifié en 4.1.6.1 pour date-time. Les caractères génériques sont autorisés dans season_start. Si tous les champs sont des caractères génériques, la saison ne commencera jamais. L'utilisation de caractères génériques exige un soin particulier pour éviter tout paramétrage conflictuel, c'est-à-dire éviter que deux saisons différentes aient le même season_start;

NOTE La saison en cours est terminée par le season_start de la saison suivante.

– week_name définit le week_profile actif dans cette saison.

4.5.5.2.9 week_profile_table_passive

Contient un tableau de week_profiles à utiliser dans les différentes saisons. Pour chaque week_profile, le day_profile pour chaque jour d'une semaine est identifié.

```
array        week_profile

week_profile::= structure
{
week_profile_name:    octet-string,
monday:               day_id,
tuesday:              day_id,
wednesday:            day_id,
thursday:             day_id,
friday:               day_id,
saturday:             day_id,
sunday:               day_id
}

day_id: unsigned
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Où:

– week_profile_name est un nom défini par l'utilisateur qui identifie le week_profile courant;

– Monday définit le day_profile valide le lundi;

– …

Sunday définit le day_profile valide le dimanche.

4.5.5.2.10 day_profile_table_passive

Contient un tableau de day_profiles, identifiés par leur day_id. Pour chaque day_profile, une liste d'actions programmées est définie par un script à exécuter et l'heure d'activation (start_time) correspondante. La liste est triée selon start_time.

```
array        day_profile

day_profile::= structure
{
day_id:              unsigned,
day_schedule:    array day_profile_action
}

day_profile_action::= structure
{
start_time:                 octet-string,
script_logical_name:    octet-string,
script_selector:            long-unsigned
}
```

Où:

– day_id est un identifiant défini par l'utilisateur, qui identifie le day_profile courant;

– start_time: définit l'heure à laquelle le script est à exécuter (aucun caractère générique). Le format suit les règles établies en 4.1.6.1 pour *time*;

– script_logical_name: définit le *logical_name* de l'objet "Script table";

– script_selector: définit le script_identifier du script à exécuter.

4.5.5.2.11 activate_passive_calendar_time

Définit l'heure à laquelle l'objet lui-même appelle la méthode spécifique *activate_passive_calendar*. Une définition avec une notation "not specified" (non spécifié) dans tous les champs de l'attribut désactivera cet automatisme. Une notation "not specified" partielle, c'est-à-dire juste dans quelques champs de la date et de l'heure, n'est pas autorisée.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date-time*.

4.5.5.3 Description de la méthode

4.5.5.3.1 activate_passive_calendar (data)

Cette méthode copie tous les attributs appelés …_passive dans les attributs correspondants appelés …_active.

```
data::= integer (0)
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.6 Register monitor (class_id = 21, version = 0)

4.5.6.1 Vue d'ensemble

Cette IC permet de modéliser la fonction de surveillance de valeurs modélisées par les objets "Data", "Register", "Extended register" ou "Demand register". Elle permet de spécifier des seuils, la valeur surveillée et un jeu de scripts (voir 4.5.2) qui sont exécutés lorsque la valeur surveillée franchit un seuil.

L'IC "Register monitor" exige une instanciation de l'IC "Script table" dans le même dispositif logique.

Register monitor		**0...n**	**class_id = 21, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. thresholds	(static)	array				x + 0x08
3. monitored_value	(static)	value_definition				x + 0x10
4. actions	(static)	array			0	x + 0x18
Méthodes spécifiques		***o/f***				

4.5.6.2 Description d'attribut

4.5.6.2.1 logical_name

Identifie l'instance de l'objet "Register monitor". Voir 6.2.13 et 6.3.10.

4.5.6.2.2 thresholds

Fournit les valeurs de seuil auxquelles l'attribut du registre référencé est comparé.

array threshold

threshold: Le seuil est du même type que l'attribut surveillé de l'objet référencé.

4.5.6.2.3 monitored_value

Définit quel attribut d'un objet est à surveiller. Seules sont autorisées les valeurs ayant des types de données simples.

```
value_definition::= structure
{
class_id:              long-unsigned,
logical_name:          octet-string,
attribute_index:       integer
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.6.2.4 actions

Définit les scripts à exécuter lorsque l'attribut surveillé de l'objet référencé franchit le seuil ("threshold") correspondant. L'attribut *actions* a exactement le même nombre d'éléments que l'attribut *thresholds*. L'ordre de classement des action_items correspond à l'ordre de classement des *thresholds* (voir 4.5.6.2.2).

```
array        action_set
action_set::=    structure
{
action_up:       action_item,
action_down:     action_item
}
```

Où:

– action_up définit l'action lorsque la valeur d'attribut du registre surveillé franchit le seuil dans le sens montant;

– action_down définit l'action lorsque la valeur d'attribut du registre surveillé franchit le seuil dans le sens descendant.

```
action_item::= structure
{
script_logical_name:    octet-string,
script_selector:        long-unsigned
}
```

4.5.7 Single action schedule (class_id = 22, version = 0)

4.5.7.1 Vue d'ensemble

Cette IC permet de modéliser l'exécution d'actions périodiques au sein d'un compteur. Ces actions ne sont pas nécessairement liées à la tarification (voir "Activity calendar" (4.5.5) ou "Schedule"(4.5.3)).

Single action schedule		**0...n**	**class_id = 22, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. executed_script	(static)	script				x + 0x08
3. type	(static)	enum				x + 0x10
4. execution_time	(static)	array				x + 0x18
Méthodes spécifiques		***o/f***				

4.5.7.2 Description d'attribut

4.5.7.2.1 logical_name

Identifie l'instance de l'objet "Single action schedule". Voir 6.2.12.

4.5.7.2.2 executed_script

Contient le nom logique de l'objet "Script table" et le sélecteur de script du script à exécuter.

```
script::= structure
{
script_logical_name:    octet-string,
script_selector:        long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Script_logical_name et script_selector définissent le script à exécuter.

4.5.7.2.3 type

enum:

(1) taille de execution_time = 1; caractère générique autorisé dans date;
(2) taille de execution_time = n; toutes les valeurs de time sont les mêmes, les caractères génériques sont interdits dans date;
(3) taille de execution_time = n; toutes les valeurs de time sont les mêmes, les caractères génériques sont autorisés dans date;
(4) taille de execution_time = n; les valeurs de time peuvent être différentes, les caractères génériques sont interdits dans date;
(5) taille de execution_time = n; les valeurs de time peuvent être différentes, les caractères génériques sont autorisés dans date.

4.5.7.2.4 execution_time

Spécifie l'heure et la date auxquelles le script est exécuté.

```
array        execution_time_date

execution_time_date::= structure
{
time: octet-string,
date: octet-string
}
```

Les deux octet-strings contiennent *time* et *date* dans cet ordre; *time* et *date* sont mises en forme comme spécifié en 4.1.6.1. Les centièmes de seconde doivent être à zéro.

4.5.8 Disconnect control (class_id = 70, version = 0)

4.5.8.1 Vue d'ensemble

Les instances de l'IC "Disconnect control" gèrent une unité de déconnexion interne ou externe du compteur (disjoncteur ou robinet de gaz, par exemple) afin de brancher ou débrancher (partiellement ou totalement) les locaux du consommateur à/de l'alimentation. Le diagramme des transitions d'état et les transitions d'état possibles sont représentés à la Figure 18.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

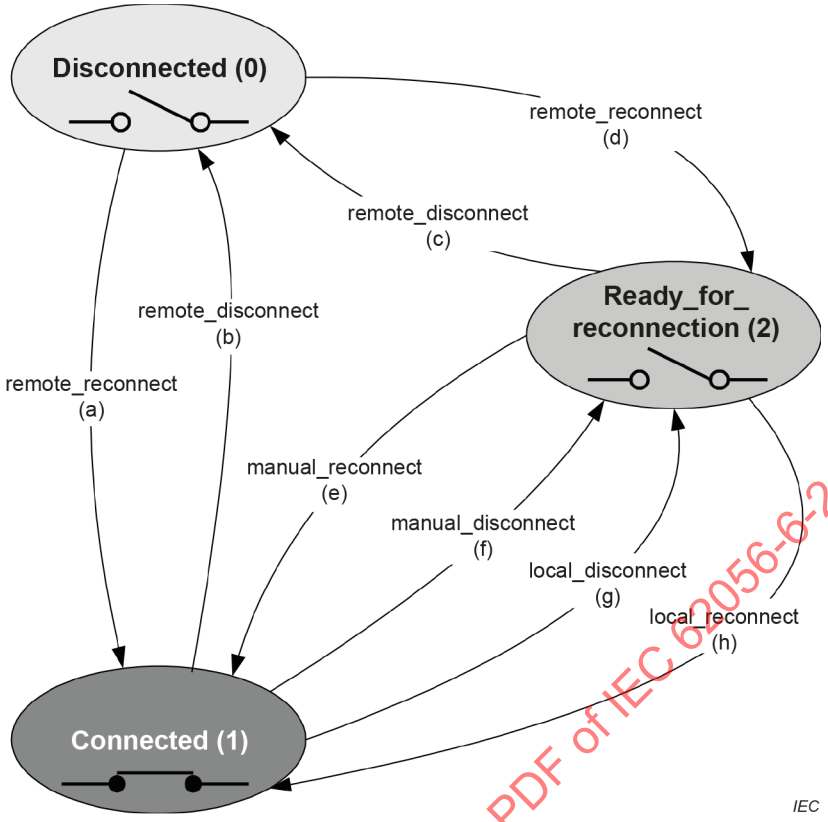


Figure 18 – Diagramme des transitions d'état de l'IC "Disconnect control"

Une déconnexion et une reconnexion peuvent être demandées:

- à distance, via une voie de communication: remote_disconnect, remote_reconnect;
- manuellement, à l'aide d'un bouton-poussoir par exemple: manual_disconnect, manual_reconnect;
- localement, par une fonction du compteur, par exemple limiteur, prépaiement: local_disconnect, local_reconnect.

Les états et les transitions d'état de l'IC "Disconnect control" sont présentés dans le Tableau 31. Les transitions d'état possibles dépendent du mode de commande. L'objet "Disconnect control" ne comporte pas de mémoire, c'est-à-dire que toute commande est exécutée immédiatement.

Afin de définir le comportement de l'objet "Disconnect control" pour chaque déclencheur, le mode de commande doit être configuré.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 31 – IC "Disconnect control" – États et transitions d'état

États		
Numéro d'état	**Nom d'état**	**Description de l'état**
0	Disconnected	L'*output_state* est mis à FALSE et le consommateur est déconnecté.
1	Connected	L'*output_state* est mis à TRUE et le consommateur est connecté.
2	Ready_for_reconnection	L'*output_state* est mis à FALSE et le consommateur est déconnecté.
Transitions d'état		
Transition	**Nom de transition**	**Description de l'état**
a	remote_reconnect	Fait passer directement l'objet "Disconnect control" de l'état Disconnected (0) à l'état Connected (1) sans intervention manuelle.
b	remote_disconnect	Fait passer l'objet "Disconnect control" de l'état Connected (1) à l'état Disconnected (0).
c	remote_disconnect	Fait passer l'objet "Disconnect control" de l'état Ready_for_reconnection (2) à l'état Disconnected (0).
d	remote_reconnect	Fait passer l'objet "Disconnect control" de l'état Disconnected (0) à l'état Ready_for_reconnection (2). À partir de cet état, il est possible de passer à l'état Connected (2) via la transition manual_reconnect (e) ou la transition local_reconnect (h).
e	manual_reconnect	Fait passer l'objet "Disconnect control" de l'état Ready_for_connection (2) à l'état Connected (1).
f	manual_disconnect	Fait passer l'objet "Disconnect control" de l'état Connected (1) à l'état Ready_for_connection (2). À partir de cet état, il est possible de revenir à l'état Connected (2) via la transition manual_reconnect (e) ou la transition local_reconnect (h).
g	local_disconnect	Fait passer l'objet "Disconnect control" de l'état Connected (1) à l'état Ready_for_connection (2). À partir de cet état, il est possible de revenir à l'état Connected (2) via la transition manual_reconnect (e) ou la transition local_reconnect (h). NOTE 1 Les transitions f) et g) sont sensiblement identiques, mais leur déclencheur est différent.
h	local_reconnect	Fait passer l'objet "Disconnect control" de l'état Ready_for_connection (2) à l'état Connected (1). NOTE 2 Les transitions e) et h) sont sensiblement identiques, mais leur déclencheur est différent.

Disconnect control		**0…n**	**class_id = 70, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	***Nom court***
1. logical_name	(static)	octet-string				x
2. output_state	(dyn.)	boolean				x + 0x08
3. control_state	(dyn.)	enum				x + 0x10
4. control_mode	(static)	enum				x + 0x18
Méthodes spécifiques		**o/f**				
1. remote_disconnect (data)		o				x + 0x20
2. remote_reconnect (data)		o				x + 0x28

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.8.2 Description d'attribut

4.5.8.2.1 logical_name

Identifie l'instance de l'objet "Disconnect control". Voir 6.2.22 et 6.2.46.

4.5.8.2.2 output_state

Indique l'état physique réel de la connexion du dispositif à l'alimentation.

boolean: TRUE = (BOOLEAN 1) Connecté,
FALSE = (BOOLEAN 0) Déconnecté

NOTE 1 En comptage d'électricité, l'alimentation est connectée lorsque le dispositif de déconnexion est fermé.

NOTE 2 En comptage de gaz et d'eau, l'alimentation est connectée lorsque le robinet est ouvert.

4.5.8.2.3 control_state

Indique l'état interne de l'objet "Disconnect control".

enum:

(0) Disconnected,
(1) Connected,
(2) Ready_for_reconnection

4.5.8.2.4 control_mode

Configure le comportement de l'objet "Disconnect control" pour tous les déclencheurs, c'est-à-dire les **transitions** d'état possibles.

control_mode	Déconnexion				Reconnexion			
	Distante		Manuelle	Locale	Distante		Manuelle	Locale
enum:	(b)	(c)	(f)	(g)	(a)	(d)	(e)	(h)
(0)	–	–	–	–	–	–	–	–
(1)	x	x	x	x	–	x	x	–
(2)	x	x	x	x	x	–	x	–
(3)	x	x	–	x	–	x	x	–
(4)	x	x	–	x	x	–	x	–
(5)	x	x	x	x	–	x	x	x
(6)	x	x	–	X	–	x	x	x
En mode (0), l'objet "Disconnect control" est toujours à l'état "connected" (connecté). La déconnexion locale est toujours possible, sauf si le déclencheur correspondant est inhibé.								

4.5.8.3 Description de la méthode

4.5.8.3.1 remote_disconnect (data)

Force l'objet "Disconnect control" à l'état "disconnected" (déconnecté) si la déconnexion distante est activée (control_mode > 0).

data::= integer (0)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.8.3.2 remote_reconnect (data)

Force l'objet "Disconnect control" à l'état "ready_for_reconnection" si une reconnexion distante directe est désactivée (control_mode = 1, 3, 5, 6).

Force l'objet "Disconnect control" à l'état "connected" si une reconnexion distante directe est activée (control_mode = 2, 4).

data::= integer (0)

4.5.9 Limiter (class_id = 71, version = 0)

4.5.9.1 Vue d'ensemble

Les instances de l'IC "Limiter" permettent de définir un ensemble d'actions qui sont exécutées lorsque la valeur d'un attribut *value* d'un objet surveillé "Data", "Register", "Extended Register", "Demand Register", etc. franchit la valeur seuil pendant au moins la durée minimale.

La valeur seuil peut être un seuil normal ou un seuil d'urgence. Le *seuil d'urgence* est activé via l'*emergency_profile* défini par l'*id de profil d'urgence*, le *temps d'activation d'urgence* et la *durée d'urgence*. L'élément *emergency profile id* (id de profil d'urgence) est mis en concordance avec un *emergency profile group id* (id de groupe de profils d'urgence): ce mécanisme permet l'activation du seuil d'urgence uniquement pour un groupe d'urgences spécifique.

Limiter		**0...n**	**class_id = 71, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	***Nom court***
1. logical_name	(static)	octet-string				x
2. monitored_value	(static)	value_definition				x + 0x08
3. threshold_active	(dyn.)	threshold				x + 0x10
4. threshold_normal	(static)	threshold				x + 0x18
5. threshold_emergency	(static)	threshold				x + 0x20
6. min_over_threshold_duration	(static)	double-long-unsigned				x + 0x28
7. min_under_threshold_duration	(static)	double-long-unsigned				x + 0x30
8. emergency_profile	(static)	emergency_profile				x + 0x38
9. emergency_profile_group_id_list	(static)	array				x + 0x40
10. emergency_profile_active	(dyn.)	boolean				x + 0x48
11. actions	(static)	action				x + 0x50
Méthodes spécifiques		***o/f***				

4.5.9.2 Description d'attribut

4.5.9.2.1 logical_name

Identifie l'instance de l'objet "Limiter". Voir 6.2.15.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.9.2.2 monitored_value

Définit un attribut d'un objet à surveiller. Seuls sont autorisés les attributs ayant des types de données simples.

```
value_definition::= structure
{
class_id:          long-unsigned,
logical_name:      octet-string,
attribute_index:   integer
}
```

4.5.9.2.3 threshold_active

Fournit la valeur de seuil active à laquelle l'attribut surveillé est comparé.

threshold: le seuil est du même type que l'attribut surveillé.

4.5.9.2.4 threshold_normal

Fournit la valeur de seuil à laquelle l'attribut surveillé est comparé en fonctionnement normal.

threshold: voir 4.5.9.2.3.

4.5.9.2.5 threshold_emergency

Fournit la valeur de seuil à laquelle l'attribut surveillé est comparé lorsqu'un profil d'urgence est actif.

threshold: voir 4.5.9.2.3.

4.5.9.2.6 min_over_threshold_duration

Définit la durée minimale au-dessus du seuil, en secondes, exigée pour exécuter l'action au-dessus du seuil.

4.5.9.2.7 min_under_threshold_duration

Définit la durée minimale en dessous du seuil, en secondes, exigée pour exécuter l'action en dessous du seuil.

4.5.9.2.8 emergency_profile

Un *emergency_profile* est défini par trois éléments: emergency_profile_id, emergency_activation_time, emergency_duration.

Un profil d'urgence est activé si l'élément emergency_profile_id concorde avec l'un des éléments de l'emergency_profile _group_id_list, et si le temps concorde avec l'emergency_activation_time et l'élément emergency_duration:

```
emergency_profile::= structure
{
emergency_profile_id:          long-unsigned,
emergency_activation_time:     octet-string,
emergency_duration:            double-long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Où:

– l'élément emergency_activation_time définit la date et l'heure auxquelles l'emergency_profile s'est activé. L'octet-string est codé comme spécifié en 4.1.6.1 pour *date-time*;

– l'élément emergency_duration définit la durée, en secondes, pendant laquelle l'*emergency_profile* est activé.

Lorsqu'un profil d'urgence est actif, l'attribut *emergency_profile_active* est mis à TRUE.

4.5.9.2.9 emergency_profile_group_id_list

Définit une liste des id de groupe du profil d'urgence.

Le profil d'urgence peut être activé uniquement si l'élément emergency_profile_id de l'attribut *emergency_profile* concorde avec l'un des éléments de l'*emergency_profile_group_id_list*:

```
array        emergency_profile_group_id

emergency_profile_group_id::= long-unsigned
```

4.5.9.2.10 emergency_profile_active

Indique que l'emergency_profile est actif.

4.5.9.2.11 actions

Définit les scripts à exécuter lorsque la valeur surveillée franchit le seuil pendant la durée minimale.

```
action::= structure
{
action_over_threshold:      action_item,
action_under_threshold:     action_item
}
```

Où:

– action_over_threshold définit l'action lorsque la valeur de l'attribut surveillé franchit le seuil dans le sens montant et reste au-dessus du seuil pendant la durée minimale au-dessus du seuil;

– action_under_threshold définit l'action lorsque la valeur de l'attribut surveillé franchit le seuil dans le sens descendant et reste en dessous du seuil pendant la durée minimale en dessous du seuil.

```
action_item::= structure
{
script_logical_name:    octet-string,
script_selector:        long-unsigned
}
```

4.5.10 Parameter monitor (class_id = 65, version = 1)

4.5.10.1 Généralités

Les instances de l'IC "Parameter monitor" sont utilisées pour surveiller une liste d'attributs d'objet COSEM qui contiennent des paramètres.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si la valeur d'un attribut de la *parameter_list* change, l'identifiant et la valeur de cet attribut sont automatiquement capturés dans l'attribut *changed_parameter*. L'heure de la modification du paramètre est capturée dans l'attribut *capture_time*. Ces attributs peuvent être capturés par un objet "Profile generic". De cette manière, un journal de toutes les modifications de paramètres peut être généré. Pour le code OBIS des objets "Parameter monitor log", voir IEC 62056-6-1:2023, 6.5.

NOTE 1 En cas de modifications de paramètre simultanées ou quasi simultanées, les modifications de l'ordre de capture et de consignation des paramètres modifiés sont à gérer par l'application.

Plusieurs objets "Parameter monitor" et objets "Profile generic" correspondants peuvent être instanciés afin de gérer un certain nombre de groupes de paramètres. Le lien entre l'objet "Parameter monitor" et l'objet "Profile generic" correspondant est assuré par l'intermédiaire de l'attribut *capture_object* de l'objet "Profile generic".

NOTE 2 Les différents paramètres pouvant être de type et de longueur différents, les entrées de la colonne du profil contenant les paramètres le sont également. Cela peut être géré en capturant différents types de paramètres dans différents objets "Profile generic" et journaux de paramètres d'une autre liste de paramètres.

NOTE 3 L'objet "Profile generic" contenant le journal des modifications de paramètres peut capturer d'autres attributs d'objet pertinents, tel que l'attribut *time* de l'objet "Clock", et d'autres valeurs appropriées.

– Utilisation de l'IC "Parameter monitor" pour détecter les modifications des paramètres de configuration:

Chaque client ayant accès au serveur doit connaître le paramétrage actuel afin de pouvoir échanger correctement des données avec des serveurs. Les paramètres peuvent être regroupés et chaque groupe peut avoir un nom.

Bien que le paramétrage puisse être connu au départ, il peut changer au cours de la durée de vie du compteur; il peut par exemple être changé par un autre client tel qu'un dispositif de service de terrain.

La configuration actuelle peut toujours être récupérée en lisant les paramètres de configuration au début de l'échange. Ceci n'est pas efficace et pas pratique en cas d'opération Push.

Une solution est nécessaire pour permettre aux clients de vérifier si la configuration du serveur est conforme aux attentes ou pour détecter si une modification a eu lieu.

À cet effet, la version 1 de l'IC "Parameter monitor" spécifie de nouveaux attributs destinés à contenir:

- le nom de la liste de paramètres (parameter_list_name);
- l'identifiant de l'algorithme de hachage (hash_algorithm_id);
- le condensé des valeurs de paramètres (parameter_value_digest);
- les valeurs des paramètres.

Lorsqu'un client configure le serveur, il peut récupérer ces attributs et les partager avec d'autres clients.

Lorsqu'un client lance un échange de données avec le serveur, il peut lire l'attribut parameter_value_digest et le comparer à la valeur qu'il a stockée. Si elles concordent, la configuration est conforme aux attentes. Sinon, le client peut lire l'attribut parameter_values pour identifier les valeurs qui ont changé.

Les attributs peuvent également être capturés dans les attributs *buffer* de l'objet "Profile generic" afin de s'assurer de la validité des données capturées. Le surcoût résultant peut être réduit en utilisant un codage "null-data".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Parameter monitor	0...n	class_id = 65, version = 1			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. changed_parameter	structure				x + 0x08
3. capture_time	date-time				x + 0x10
4. Parameter_list	array				x + 0x18
5. parameter_list_name	octet-string				x + 0x20
6. hash_algorithm_id	enum				x + 0x28
7. parameter_value_digest	octet-string				x + 0x30
8. parameter_values	structure				x + 0x38
Méthodes spécifiques	***o/f***				x + 0x40
1. add_parameter (data)	f				x + 0x48
2. delete_parameter (data)	f				x + 0x50

4.5.10.2 Description d'attribut

4.5.10.2.1 logical_name

Identifie l'instance de l'objet "Parameter monitor". Voir 6.2.14.

4.5.10.2.2 changed_parameter

Contient l'identifiant et la valeur du dernier paramètre modifié.

```
structure
{
class_id:          long-unsigned,
logical_name:      octet_string,
attribute_index:   integer,
attribute_value:   CHOICE
-- CHOICE comme indiqué dans la classe d'interfaces "Data"
}
```

4.5.10.2.3 capture_time

Fournit les informations de date et heure de capture de la valeur de l'attribut *changed_parameter*.

date-time est mis en forme comme spécifié en 4.1.6.1.

4.5.10.2.4 parameter_list

Contient la liste des paramètres gérés par une instance donnée de l'IC "Parameter monitor".

```
parameter_list::= array parameter_list_element

parameter_list_element::= structure
{
class_id:          long-unsigned,
logical_name:      octet_string,
attribute_index:   integer
}
```

NOTE La liste des paramètres surveillés peut être modifiée en utilisant la méthode *add_parameter* ou *delete_parameter* ou en écrivant cet attribut.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.10.2.5 parameter_list_name

Contient le nom donné à la liste de paramètres définie par l'attribut parameter_list.

4.5.10.2.6 hash_algorithm_id

enum

(0) SHA-256,
(1) SHA-384,
(2) Les 16 derniers octets de SHA-256,
(3) Les 8 derniers octets de SHA-256,
(4) Les 4 derniers octets de SHA-256.

4.5.10.2.7 parameter_value_digest

Résultat du calcul de condensé selon l'algorithme spécifié par l'attribut hash_algorithm_id. Le condensé est calculé sur toutes les valeurs de la liste de paramètres définie par l'attribut parameter_list dans l'ordre où elles sont placées dans le tableau. Pour calculer le condensé, l'ensemble complet des valeurs de l'attribut parameter_list est d'abord transformé en chaîne d'octets. Les règles applicables pour cette transformation sont définies dans l'IEC 62056-5-3:2023, 5.3.4.3. Une conversion de données est effectuée pour tous les types de données qui ne sont pas une chaîne d'octets.

4.5.10.2.8 parameter_values

Contient une copie des valeurs des attributs référencés dans l'attribut parameter_list.

Il s'agit d'une structure qui contient la valeur codée A-XDR de chaque attribut dans l'ordre du tableau parameter_list.

4.5.10.3 Description de la méthode

4.5.10.3.1 add_parameter (data)

Ajoute un paramètre à *parameter_list*.

data::= parameter_list_element

NOTE Un paramètre peut être consigné dès qu'il a été ajouté à la liste. L'ajout d'un élément à la liste n'a aucun impact sur le *buffer* de l'objet "Profile generic" qui capture l'attribut *changed_parameter*.

4.5.10.3.2 delete_parameter (data)

Supprime un paramètre de *parameter_list*.

data::= parameter_list_element

NOTE Si un paramètre est supprimé de la liste de paramètres, ses modifications ne sont plus journalisées. La suppression d'un élément de la liste n'a aucun impact sur le *buffer* de l'objet "Profile generic" qui capture l'attribut *changed_parameter*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11 Sensor manager (class_id = 67, version = 0)

4.5.11.1 Généralités

NOTE Cette classe d'interfaces est principalement utilisée pour le comptage des types d'énergie autres que l'électricité.

La plupart des instruments de mesure relevant du domaine d'application de la MID fonctionnent avec des capteurs dédiés (transducteurs et émetteurs) reliés à l'unité de traitement. Le fonctionnement et les limites de ces capteurs doivent être surveillés en permanence afin de s'assurer qu'ils satisfont aux exigences métrologiques en vue d'un calcul ultérieur de valeurs monétaires.

De plus, les valeurs mesurées doivent être surveillées. Ces valeurs peuvent être reliées à une grandeur physique – valeurs brutes de tension, d'intensité, de résistance, de fréquence, de sortie numérique – fournie par le capteur et aux grandeurs mesurées résultant du traitement des informations fournies par le capteur.

Il est nécessaire de surveiller et de journaliser souvent les valeurs pertinentes afin d'obtenir des informations de diagnostic qui permettent:

- l'identification du dispositif capteur;
- la connexion et l'état de scellé du capteur;
- la configuration des capteurs;
- la surveillance du fonctionnement des capteurs;
- la surveillance du résultat du traitement.

La classe d'interfaces "Sensor manager" permet de gérer les informations détaillées associées à un capteur au moyen d'un seul objet.

Pour des capteurs/dispositifs plus simples, les objets COSEM déjà existants – identifiant les capteurs, contenant les valeurs de mesure et surveillant ces valeurs de mesure – peuvent être utilisés.

4.5.11.2 Vue d'ensemble

Les instances de l'IC "Sensor manager" gèrent des informations complexes relatives aux capteurs. Elles permettent également de surveiller les données brutes et la valeur traitée, obtenue par traitement des données brutes à l'aide d'algorithmes appropriés exigés par l'application particulière. Cette IC inclut un certain nombre de fonctions:

- données de plaque signalétique du capteur et informations de site (attributs 2 à 6);
- une fonction "Extended register" pour la valeur brute (*raw-value*) (attributs 7 à 10);
- une fonction "Register monitor" pour la valeur brute (*raw-value*) (attributs 11-12);

  NOTE 1 Toutes les données brutes (la tension de sortie d'un capteur de pression, par exemple) n'ont pas leur propre code/objet OBIS. C'est la raison pour laquelle les données brutes sont incluses dans la classe "Sensor manager".

- une fonction "Register monitor" pour la valeur traitée (*processed_value)* (attributs 13 à 15).

  NOTE 2 Tous les "modules" ne sont pas nécessairement présents. Les attributs non utilisés peuvent ne pas être mis en œuvre ou ne pas être accessibles.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Sensor manager		0...n	class_id = 67, version = 0			
Attribut(s)		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. serial_number	(dyn.)	octet-string				x + 0x08
3. metrological_identification	(dyn.)	octet-string				x + 0x10
4. output_type	(dyn.)	enum				x + 0x18
5. adjustment_method	(dyn.)	octet-string				x + 0x20
6. sealing_method	(dyn.)	enum				x + 0x28
7. raw_value	(dyn.)	CHOICE				x + 0x30
8. scaler_unit	(dyn.)	structure				x + 0x38
9. status	(dyn.)	CHOICE				x + 0x40
10. capture_time	(dyn.)	date-time				x + 0x48
11. raw_value_thresholds	(dyn.)	array				x + 0x50
12. raw_value_actions	(dyn.)	array				x + 0x58
13. processed_value	(dyn.)	processed_value_ definition				x + 0x60
14. processed_value_thresholds	(dyn.)	array				x + 0x68
15. processed_value_actions	(dyn.)	array				x + 0x70
Méthodes spécifiques		***o/f***				
1. reset (data)		*f*				x + 0x80

4.5.11.3 Description d'attribut

4.5.11.3.1 logical_name

Identifie l'instance de l'objet "Sensor manager".

NOTE Les objets "Sensor manager" sont utilisés en relation avec la mesure non électrique. Par conséquent, aucun code OBIS n'est défini dans le présent document.

4.5.11.3.2 serial_number

Identifie le capteur (->informations de site).

NOTE Pour un capteur simple, le numéro de série peut être détenu par les objets "Data" avec des codes OBIS appropriés s'il s'agit de la seule information exigée.

4.5.11.3.3 metrological_identification

Décrit les informations du capteur pertinentes du point de vue métrologique, par exemple l'identifiant métrologique ou la date d'étalonnage.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11.3.4 output_type

Décrit la sortie physique du capteur (->informations de site)

enum:

(0) non spécifié,
(1) 4-20 mA,
(2) 0-20 mA,
(3) 0-5 V,
(4) 0-10 V,
(5) Pt100,
(6) Pt500,
(7) Pt1000,
(8) HART,
(128) spécifique au fabricant
Toutes les autres valeurs sont réservées.

4.5.11.3.5 adjustment_method

Décrit la méthode d'ajustement du capteur, par exemple par l'équation à trois points de mesure.

4.5.11.3.6 sealing_method

Type de scellés appliqués au capteur.

enum:

(0) aucun;
(1) mécanique (fil métallique ou étiquette de protection, par exemple);
(2) électronique (contact, par exemple);
(3) logiciel (protection par mot de passe, par exemple)

4.5.11.3.7 raw_value

Valeur physique provenant du capteur (tension provenant d'un convertisseur de pression en tension, par exemple).

Pour les choix possibles de types de données, voir IC "Register" (4.3.3.2.2).

4.5.11.3.8 scaler_unit

Échelle et unité de *raw_value*.

Pour la définition de *scaler_unit*, voir IC "Register" (4.3.3.2.3).

4.5.11.3.9 status

État de la dernière *raw_value* capturée (pour la définition de *status*, voir IC "Extended register" [4.3.3.2.4]). *status* conserve des informations telles que:

– la défaillance d'un capteur;
– un capteur activé/non activé.

La signification des éléments de *status* doit être fournie pour chaque instance de l'objet "Sensor manager".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11.3.10 capture_time

Fournit les informations de date et d'heure spécifiques à l'objet "Sensor manager" indiquant le moment auquel la valeur de l'attribut *raw_value* a été capturée.

date_time est mis en forme comme spécifié en 4.1.6.1.

Pour les choix possibles de types de données, voir classe d'interfaces "Extended register" (4.3.3.2.5).

4.5.11.3.11 raw_value_thresholds

Fournit les valeurs de seuil auxquelles l'attribut *raw_value* est comparé.

array threshold

threshold: Le seuil ("threshold") est du même type que les données brutes ("raw-data") (4.5.11.3.7).

4.5.11.3.12 raw_value_actions

Définit les scripts à exécuter lorsque la *raw_value* franchit le seuil correspondant. L'attribut *raw_value_actions* a exactement le même nombre d'éléments que l'attribut *raw_value_thresholds*. L'ordre de classement des *action_items* correspond à l'ordre de classement des seuils. Pour la spécification des actions, voir IC *Register monitor* en 4.5.6.2.4.

4.5.11.3.13 processed_value

Référence l'attribut contenant la valeur traitée des données brutes fournies par le capteur.

```
processed_value_definition::= structure
{
        class_id:       long-unsigned,
        logical_name:   octet_string,
        attribute_index: integer
}
```

A noter que plusieurs objets "Sensor manager" et objets de valeur traitée peuvent appartenir au même capteur.

4.5.11.3.14 processed_value_thresholds

Fournit les valeurs de seuil auxquelles la valeur traitée est comparée.

array threshold

threshold: Le seuil est du même type que la valeur traitée (référencée par l'attribut *processed-value*) (4.5.11.3.13).

4.5.11.3.15 processed_value_actions

Définit les scripts à exécuter lorsque la valeur traitée franchit le seuil correspondant.

L'attribut *processed_value_actions* a exactement le même nombre d'éléments que l'attribut *processed_value_thresholds*. L'ordre de classement des *action_items* correspond à l'ordre de classement des seuils.

Pour la spécification des actions, voir IC *Register monitor* en 4.5.6.2.4.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.11.4 Description de la méthode

4.5.11.4.1 reset (data)

Cette méthode réinitialise la *raw_value* à la valeur par défaut. La valeur par défaut est une constante spécifique à l'instance.

L'attribut *capture_time* est défini à l'heure de l'exécution de la réinitialisation.

data::= integer (0)

4.5.11.5 Exemple pour un capteur de pression absolue

La Figure 19 illustre la définition des seuils supérieur et inférieur applicables.

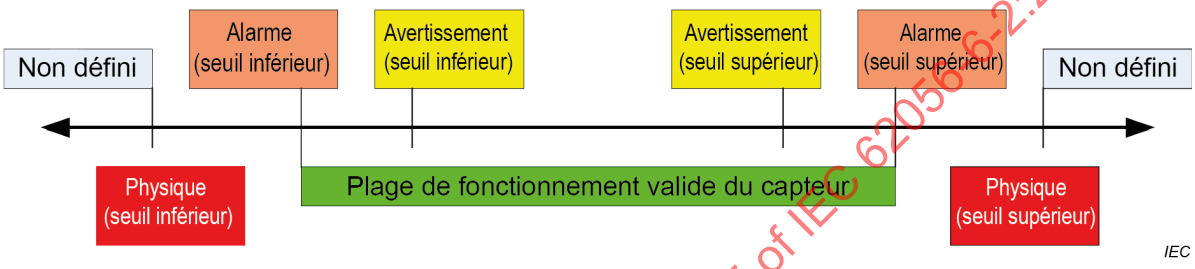


Figure 19 – Définition des seuils supérieurs et inférieurs

Le Tableau 32 et le Tableau 33 donnent des exemples de différents seuils et des actions à mener lorsque les seuils sont franchis.

Tableau 32 – Présentation explicite des tableaux de valeurs de seuils

Threshold (seuil)	Physique (seuil inférieur)	Physique (seuil supérieur)	Alarme (seuil inférieur)	Alarme (seuil supérieur)	Avertissement (seuil inférieur)	Avertissement (seuil supérieur)
Valeur	1,0	5,5	1,2	5,0	1,4	4,8
scaler_unit	1, Volt	1, Volt	1, bar	1, bar	1, bar	1, bar

Tableau 33 – Présentation explicite d'action_sets

action_set	Physique (seuil inférieur)	Physique (seuil supérieur)	Alarme (seuil inférieur)	Alarme (seuil supérieur)	Avertissement (seuil inférieur)	Avertissement (seuil supérieur)
action_up	clr_phy_alarm_bit	set_phy_alarm_bit	clear_alarm_bit	set_alarm_bit	clear_warn_bit	set_warn_bit
action_down	set_phy_alarm_bit	clr_phy_alarm_bit	set_alarm_bit	clear_alarm_bit	set_warn_bit	clear_warn_bit

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.12 Arbitrator (class_id = 68, version = 0)

4.5.12.1 Vue d'ensemble

Les instances de l'IC "Arbitrator" permettent de déterminer, selon des règles préconfigurées composées de permissions et de pondérations, l'action à réaliser lorsque plusieurs acteurs peuvent demander des actions potentiellement conflictuelles pour contrôler la même source. En règle générale, un objet "Arbitrator" est instancié pour chaque ressource dont les demandes d'actions conflictuelles doivent être traitées.

L'IC "Arbitrator" permet:

- de configurer les possibles et potentielles actions conflictuelles qui peuvent être demandées;
- de configurer les permissions accordées à chaque acteur pour demander les actions possibles;
- de configurer la pondération pour chaque acteur et pour chaque demande possible.

NOTE 1 L'interrupteur de commande d'alimentation ou un robinet de gaz du compteur sont des exemples de ressource. La déconnexion de la fourniture, l'activation de la reconnexion, la reconnexion de l'alimentation, la prévention contre les déconnexions, la prévention contre les reconnexions sont des exemples d'actions possibles.

Les actions qui peuvent être demandées sont traitées dans l'attribut *actions* sous la forme d'un tableau d'identifiants de script. Les scripts (traités par des objets "Script table" distincts) permettent d'exécuter un large éventail de fonctions. Des actions peuvent être conçues pour inhiber l'exécution d'autres actions: elles sont modélisées sous forme de scripts vides, c'est-à-dire pointant vers un script_identifier (0) d'un objet "Script table".

NOTE 2 Les objets "Arbitrator" ne contiennent pas les noms des actions, mais leur but et leurs effets peuvent être déduits en consultant les objets "Script table" pertinents.

Les permissions sont traitées dans l'attribut *permissions_table* sous la forme d'un tableau de chaînes binaires: chaque élément du tableau représente un acteur, et chaque bit de la chaîne binaire représente une action du tableau *actions*.

Les pondérations sont traitées dans l'attribut *weightings_table* sous la forme d'un tableau à deux dimensions: chaque ligne représente un acteur, et une pondération est attribuée à chaque action possible de cet acteur. Dans le cas des actions conçues pour inhiber d'autres actions, une pondération très élevée peut être attribuée comparée à celle de l'action qui doit être inhibée.

Les actions sont demandées en appelant la méthode *request_action*. Les paramètres d'appel de méthode contiennent:

- l'identifiant de l'acteur. Cet élément, un nombre sans signe, pointe vers une ligne des attributs *permissions_table, weightings_table* et *most_recent_requests_table*;

  NOTE 3 Les noms des acteurs peuvent être précisés dans des spécifications d'accompagnement spécifiques au projet.

- la liste des actions demandées, sous la forme d'une chaîne binaire. Chaque bit correspond à un élément du tableau *actions*: pour les actions demandées, la valeur 1 est attribuée au bit, et pour les actions non demandées (inactions), la valeur 0 est attribuée au bit. Un acteur peut ne rien demander, ou demander une, plusieurs ou toutes les actions dans une seule demande d'un appel. La possibilité de demander plusieurs actions dans une seule demande s'explique par le fait de permettre à l'acteur de demander une action tout en empêchant un autre acteur d'inverser cette action.

  NOTE 4 Par exemple, un acteur peut demander de débrancher l'alimentation et empêcher un autre acteur de la rebrancher.

  NOTE 5 Une demande d'action antérieure formulée par un acteur peut être annulée en ne demandant pas la même action (c'est-à-dire en demandant une inaction) dans un autre appel de la méthode *request_action* par le même acteur. Une demande d'inhibition de l'action est alors formulée.

L'attribut *most_recent_requests_table* contient la liste des demandes les plus récentes de chaque acteur, sous la forme d'un tableau de chaînes binaires: chaque élément du tableau représente la dernière demande d'un acteur, et chaque bit de la chaîne binaire représente une action/inaction demandée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si la méthode *request_action* est appelée par un acteur, l'AP exécute toutes les activités suivantes:

- il vérifie l'entrée de l'attribut *permissions_table* de l'acteur donné pour évaluer si les actions demandées sont admises ou pas;
- il met à jour l'attribut *most_recent_requests_table* en définissant ou effaçant le bit de la chaîne binaire de cet acteur pour chaque action demandée qui est également admise (bit défini) ou non demandée/non admise (bit effacé);
- il applique le *weightings_table* pour le *most_recent_requests_table*: pour chaque bit défini dans l'attribut *most_recent_requests_table*, la pondération correspondante de chaque acteur est appliquée;
- pour chaque action, les pondérations sont ajoutées; puis
- en présence d'une seule pondération totale la plus élevée pour une action, cette valeur est écrite dans l'attribut *last_outcome*, et le script correspondant est exécuté. En l'absence d'une seule pondération totale la plus élevée, rien ne se passe.

Arbitrator		**0...n**	**class_id = 68, version = 0**			
Description d'attribut		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. actions	(static)	array			toutes vides	x + 0x08
3. permissions_table	(static)	array			tous les bits effacés	x + 0x10
4. weightings_table	(static)	array			tous nuls	x + 0x18
5. most_recent_requests_table	(dyn.)	array			tous les bits effacés	x + 0x20
6. last_outcome	(dyn.)	unsigned	0	n	0	x + 0x28
Méthodes spécifiques		**o/f**				
1. request_action (data)		o				x + 0x30
2. reset (data)		f				x + 0x38

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.5.12.2 Description d'attribut

4.5.12.2.1 logical_name

Identifie l'instance de l'objet "Arbitrator". Voir 6.2.47.

4.5.12.2.2 actions

Définit les actions qui peuvent être demandées.

```
actions::= array action_item

action_item::= structure
{
script_logical_name:    octet-string,
script_selector:        long-unsigned
}
```

Où:

– script_logical_name: définit le *logical_name* de l'objet "Script table";

– script_selector: définit le script_identifier du script à exécuter.

Les entrées qui sont destinées à inhiber d'autres actions sont représentées par des scripts vides, c'est-à-dire pointant vers le script_identifier (0) d'un objet "Script table".

4.5.12.2.3 permissions_table

Contient les permissions accordées à chaque acteur pour demander les actions possibles.

permissions_table::= array actor_permissions

actor_permissions::= bit-string

Chaque entrée représente les permissions accordées à un acteur pour demander chaque action.

Chaque bit de la chaîne binaire correspond à une action du tableau *actions*. La longueur de la chaîne binaire doit être égale au nombre d'éléments du tableau *actions*.

Le premier bit et le dernier bit correspondent respectivement au premier élément et au dernier élément du tableau *actions*.

Les bits doivent être définis pour les actions admises, et effacés pour les actions non admises.

NOTE Ces *actor_permissions* modélisent les règles métier plutôt que d'agir comme un mécanisme de contrôle des accès.

4.5.12.2.4 weightings_table

Contient la pondération attribuée pour chaque acteur et à chaque action possible de cet acteur.

weightings_table::= array actor_weighting_list

actor_weighting_list::= array actor_action_weight

actor_action_weight::= long-unsigned

Le nombre et l'ordre des éléments du tableau *weightings_table* doivent être identiques à ceux du tableau *permissions_table*.

Le nombre d'éléments du tableau *actor_weighting_list* doit être identique à ceux du tableau *actions*. Le premier élément et le dernier élément indiquent respectivement la pondération pour la première action et la dernière action.

NOTE Il est préférable d'utiliser une pondération de puissance 2. L'utilisation de différentes pondérations pour différents acteurs et différentes actions permet d'éviter les situations dans lesquelles l'évaluation de la demande d'action ne donne aucun résultat unique (auquel cas aucune action n'est réalisée).

4.5.12.2.5 most_recent_requests_table

Contient les demandes les plus récentes de chaque acteur.

most_recent_requests_table::= array most_recent_request
most_recent_request::= bit-string

Chaque entrée représente la demande la plus récente d'un acteur.

Le nombre et l'ordre des éléments du tableau *most_recent_requests_table* doivent être identiques à ceux du tableau *permissions_table*.

La longueur de la chaîne binaire *most_recent_request* doit être égale au nombre d'éléments du tableau *actions*. Le premier bit et le dernier bit correspondent respectivement au premier élément et au dernier élément du tableau *actions*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Le bit est défini pour chaque action demandée et qui est également permise. Le bit est effacé pour chaque action qui n'est pas demandée (inaction) ou qui n'est pas admise.

4.5.12.2.6 last_outcome

Contient le résultat de la demande la plus récente ayant donné lieu à une seule pondération totale la plus élevée pour une action demandée et donc exécutée.

Le nombre identifie un bit de la chaîne binaire request_action_list: le nombre 1 correspond au premier bit et, par conséquent, au premier élément du tableau *actions*.

4.5.12.3 Description de la méthode

4.5.12.3.1 request_action (data)

Définit les actions demandées par un acteur.

```
data::= structure
{
        request_actor:              unsigned,
        request_action_list:        bit-string
}
```

Où:

– request_actor est un indice dans les tableaux correspondants. Le nombre 1 identifie la première entrée des tableaux *permissions_table, weightings_table* et *most_recent_requests_table*. Le nombre *n* identifie l'entrée la plus élevée de ces tableaux;

– request_action_list identifie la ou les actions à exécuter. Le bit est défini pour chaque action demandée. Le bit n'est pas défini pour chaque action non demandée (inaction). La longueur de la chaîne binaire doit être égale au nombre d'éléments du tableau *actions*. Le premier bit et le dernier bit correspondent respectivement au premier élément et au dernier élément.

Même si plusieurs actions peuvent être demandées, une seule action (modélisée par un script) sera réellement exécutée, à condition qu'elle soit permise pour cet acteur et qu'il existe un seul résultat total le plus élevé. Comme indiqué ci-dessus, une action peut être un script vide.

4.5.12.3.2 reset (data)

Efface les champs configurables de l'instance d'objet, c'est-à-dire:

– efface tous les bits de l'attribut *permissions_table*;

– met toutes les valeurs de l'attribut *weightings_table* à zéro;

– efface tous les bits de l'attribut *most_recent_requests_table*;

– met l'attribut *last_outcome* à zéro.

NOTE À la suite d'une réinitialisation, il peut être prévu que chaque appel de *request_action* laisse l'attribut *last_outcome* égal à zéro, et qu'aucune action ne soit exécutée étant donné que les éléments de l'attribut *permissions_table* sont tous effacés.

```
data::= integer (0)
```

4.5.13 Exemples de modélisation: tarification et facturation

La Figure 20 présente un exemple de modélisation du paramétrage et de la gestion des tarifs à l'aide d'objets COSEM.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

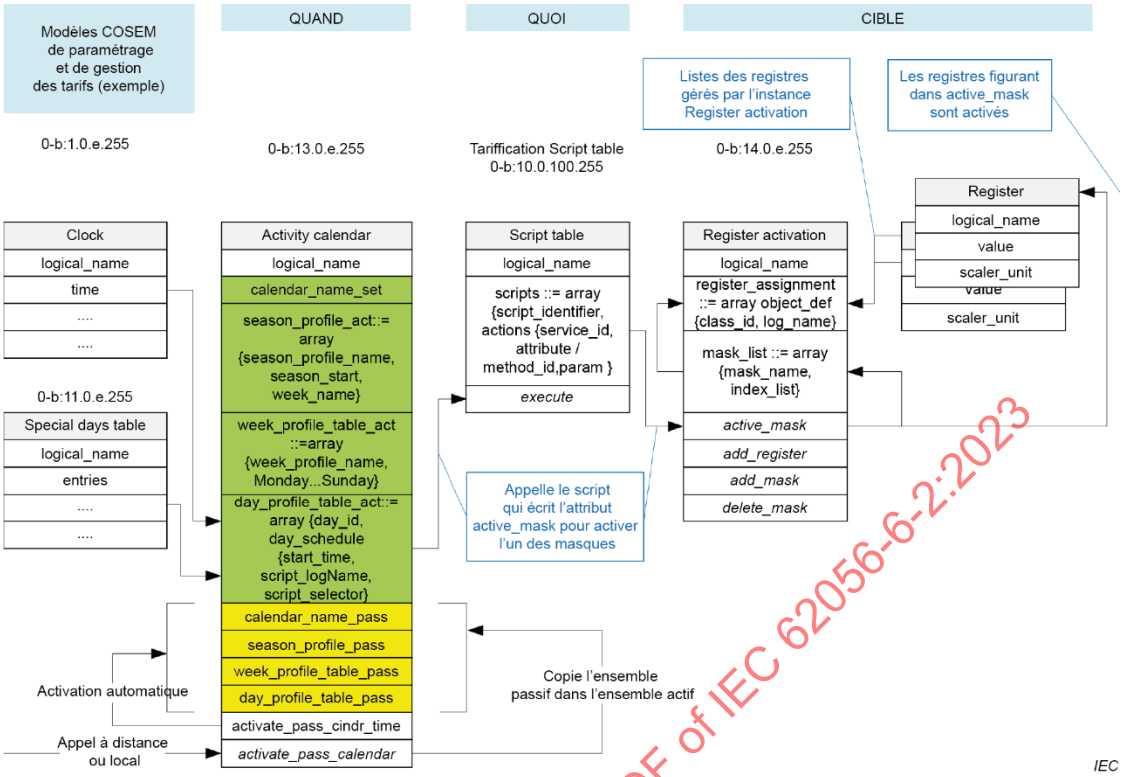


Figure 20 – Modèle de tarification COSEM (exemple)

La Figure 21 présente un exemple de modélisation du paramétrage et de la gestion de la facturation à l'aide d'objets COSEM.

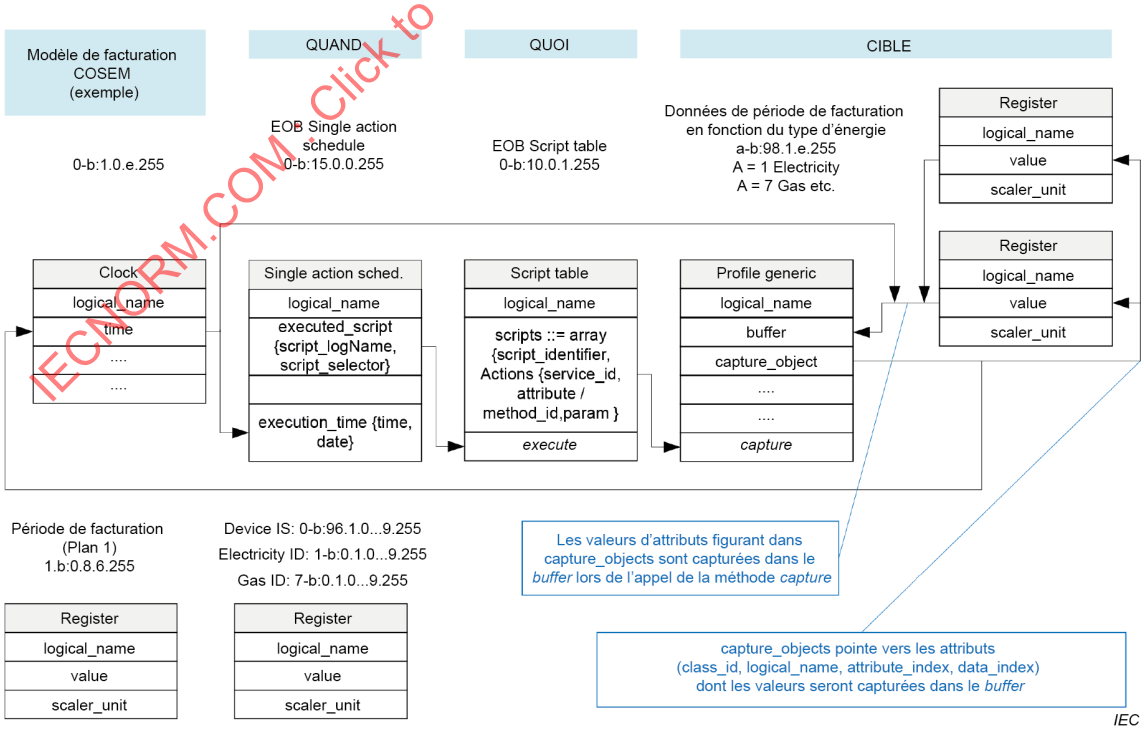


Figure 21 – Modèle de facturation COSEM (exemple)

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6 Classes d'interfaces relatives au comptage à paiement

4.6.1 Présentation du modèle comptable COSEM

Le modèle comptable COSEM contient quatre classes d'interfaces conjuguées: "Account", "Credit", "Charge" et "Token gateway". Ces classes concernent l'énergie, pas sa livraison. La classe "Account" est liée à ses objets "Credit", "Charge" et "Token gateway" par un groupe de valeurs D et un champ B, de sorte qu'il convient qu'une classe "Account" D=0 soit liée à une classe "Token gateway" D=40 et qu'elle dispose d'objets "Credit" D=10 et d'objets "Charge" D=20. Alors qu'il convient qu'une classe "Account" D=1 dispose d'objets "Token gateway" D=41, "Credit" D=11 et "Charge" D=21, etc., plusieurs objets "Token gateway", "Credit" et "Charge" associés à la même classe "Account" sont identifiés à l'aide de différentes valeurs du champ du groupe de valeurs E.

Un objet "Account" contient des informations récapitulatives et coordonne les informations relatives aux crédits et aux charges. Il n'y a qu'un seul objet "Account" par alimentation. Par exemple, l'importation d'électricité est associée à un objet "Account", mais un système qui dispose également d'une microgénération peut avoir un deuxième objet "Account" permettant de gérer l'exportation de l'électricité générée. Le deuxième objet "Account" peut être accessible ou non par l'intermédiaire de la même Association d'application (AA) que le premier.

Un objet "Credit" contient des informations détaillées relatives à une source de fonds. Au moins un objet "Credit" est (en général) associé à un objet "Account": par exemple, un objet correspondant à un crédit de jeton et un autre correspondant à un crédit d'urgence. Ces deux objets peuvent recevoir des montants de crédit provenant de jetons, mais le crédit d'urgence peut recevoir des montants de crédit uniquement lorsqu'il a été utilisé (en totalité ou en partie) et lorsque le bit 2 est attribué à *credit_configuration* (exige le remboursement du montant du crédit).

Il existe plusieurs types de crédits énumérés dans IEC TR 62055-21:2005, lesquels sont pris en charge par l'IC "Credit". Il peut exister aucune ou plusieurs instances de chaque type de crédit.

- **token_credit:** Crédit transféré vers un compteur fonctionnant en mode de prépaiement, en principe sous la forme de jetons de crédit;

  NOTE 1 Dans un compteur fonctionnant en mode de crédit ou en mode de paiement géré, un objet "Credit" configuré avec le type token_credit est utilisé pour enregistrer le montant du crédit utilisé depuis la dernière synchronisation avec un client.

  NOTE 2 Le contenu du jeton n'est pas défini par le présent document. Il peut s'agir d'une somme d'argent ou d'une autre grandeur qui peut être comptabilisée de manière équivalente à la devise utilisée par le compteur.

- **reserved_credit:** Crédit en réserve, qui est débloqué dans des conditions particulières;
- **emergency_credit:** Fonctions comptables portant sur le calcul et la négociation du crédit débloqué uniquement dans des situations d'urgence. En règle générale, le montant du crédit d'urgence utilisé est récupéré à partir du jeton de crédit ensuite acheté;

  NOTE 3 L'urgence mentionnée ci-dessus fait référence à un moment où le consommateur ne dispose pas de crédit de jeton, et non à une situation liée à la sécurité.

- **time_based_credit:** Crédit débloqué selon un calendrier précis;
- **consumption_based_credit:** Crédit débloqué selon un calendrier de niveaux de consommation. Par exemple, si un consommateur maintient son niveau de consommation au-dessous d'un certain seuil, le système peut débloquer un montant de crédit prédéfini.

Un objet "Charge" contient des informations détaillées relatives à un ensemble de dépenses, à savoir un moyen d'utiliser un crédit. Un ou (le plus souvent) plusieurs objets "Charge" peuvent être associés à un objet "Account" (un pour l'utilisation de l'énergie, un pour la redevance d'abonnement et éventuellement un pour régler une dette comme des frais d'installation, par exemple).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

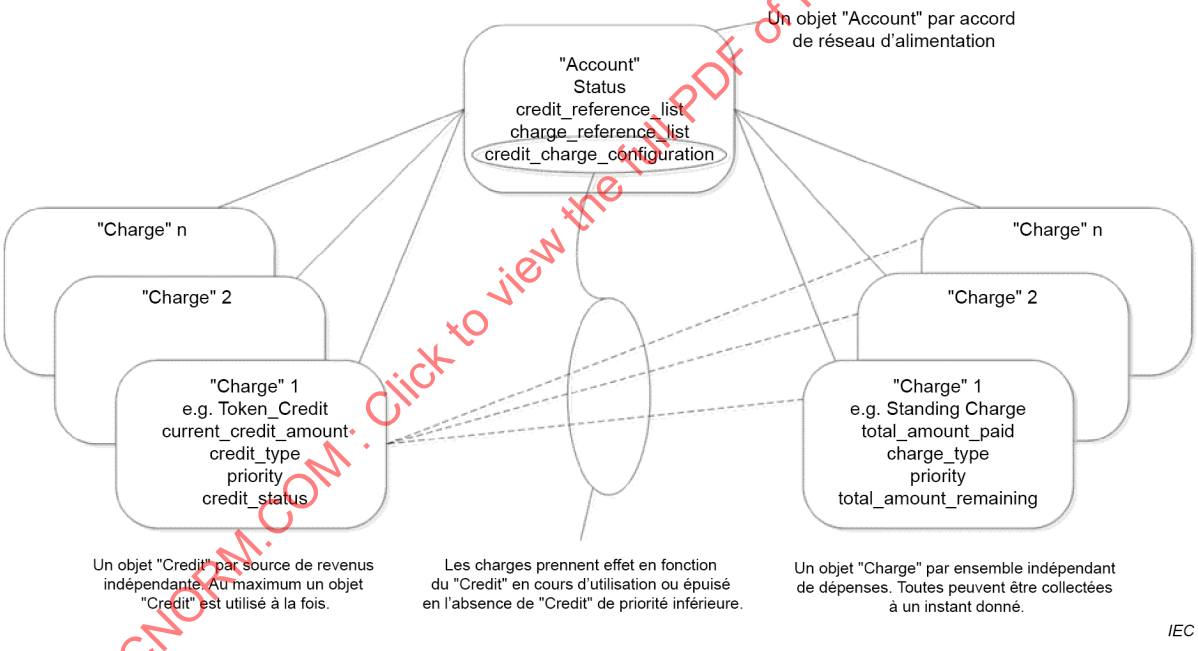
Il existe plusieurs types de charges énumérées dans l'IEC TR 62055-21:2005, mais les types suivants, qui se distinguent par leur déclencheur pour la collecte, sont les plus utiles dans le modèle comptable COSEM. Il peut exister aucune ou plusieurs instances de chaque type d'IC "Charge".

- **consumption_based_collection:** décrit les charges collectées selon le montant de la consommation dans le cadre d'un tarif. Un prix unitaire est attribué à chaque registre de tarif de l'énergie consommée;

  NOTE 4 Les tarifs ne peuvent pas être appliqués lorsque la devise est en temps ou en unités d'énergie.

- **time_based_collection:** décrit les charges collectées régulièrement dans le temps, quelle que soit la consommation au cours de cette période. Cela peut permettre de collecter les redevances d'abonnement ou de régler une créance sur une période;

- **payment_event_based_collection:** décrit les charges collectées à partir de chaque rechargement reçu, en général pour le remboursement d'une créance. Cela peut être exprimé **en fonction du montant**, un montant fixe étant prélevé de chaque crédit de rechargement reçu (le consommateur paye 2 euros sur chaque vente, quel que soit le montant de la vente, par exemple) ou ***percentage_based_collection*** , une partie du montant du crédit de rechargement reçu étant prélevée (à chaque vente, le consommateur paye 20 % du montant de la vente, par exemple). Le bit 0 (collecte selon un pourcentage) de l'attribut *charge_configuration* de l'objet "Charge" spécifie la méthode d'*event_based_collection*. La Figure 22 donne un aperçu général du modèle comptable.



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 22 – Présentation du modèle comptable

La Figure 23 présente les instances des classes d'interfaces "Account", "Credit" et "Charge" avec certains de leurs attributs, ainsi que les relations qu'entretiennent ces attributs. Dans cet exemple:

a) un objet "Account", deux objets "Credit" et deux objets "Charge" sont configurés;

b) "Credit" 1 est de type *token_credit*, les attributs *low_credit_threshold* et *limit* étant définis sur 0;

c) l'interaction entre plusieurs classes est présentée sur ce schéma. La configuration détaillée des objets "Credit" et "Charge" n'est pas présentée.

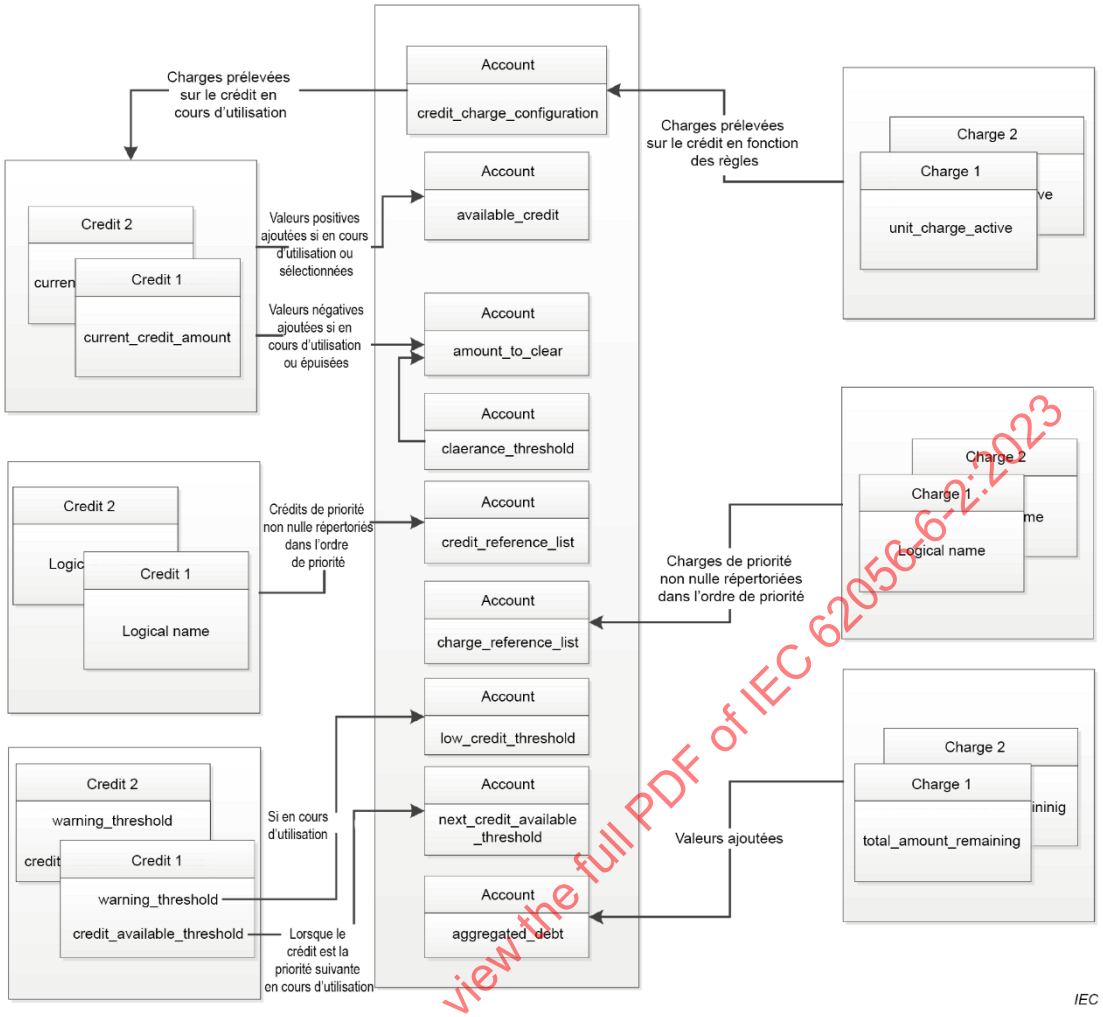


Figure 23 – Schéma des relations entre les attributs

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2 Account (class_id = 111, version = 0)

4.6.2.1 Vue d'ensemble

Les instances de l'IC "Account" gèrent tous les éléments nécessaires liés à la surveillance des objets "Credit" et des objets "Charge" référencés par une instance particulière de l'IC "Account".

Le fonctionnement du comptage à paiement sera défini à l'intérieur des objets "Charge", "Credit" et "Account" et des règles de déconnexion.

NOTE Les règles de déconnexion sont spécifiques à l'application ou peuvent être modélisées à l'aide d'une instance de l'IC "Arbitrator" (voir 4.5.12).

Si cela est spécifié de manière explicite pour un projet particulier, il est admis de passer du mode de crédit au mode de prépaiement ou inversement, une fois la configuration comptable correctement établie et l'objet "Account" activé. En l'absence de ce type de spécification, il convient que le mode de fonctionnement ne change pas pour un objet "Account" particulier une fois qu'il a été activé.

Account		0...n	class_id = 111, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string			s.o.	x
2. account_mode_and_status		structure			0, 0	x + 0x08
3. current_credit_in_use	(dyn.)	unsigned			0	x + 0x10
4. current_credit_status	(dyn.)	bit-string			Tous les bits effacés	x + 0x18
5. available_credit	(dyn.)	double-long			0	x + 0x20
6. amount_to_clear	(dyn.)	double-long			0	x + 0x28
7. clearance_threshold	(static)	double-long			0	x + 0x30
8. aggregated_debt	(dyn.)	double-long			0	x + 0x38
9. credit_reference_list	(static)	array			vide	x + 0x40
10. charge_reference_list	(static)	array			vide	x + 0x48
11. credit_charge_configuration	(static)	array			vide	x + 0x50
12. token_gateway_configuration	(static)	array			vide	x + 0x58
13. account_activation_time	(static)	octet-string			s.o.	x + 0x60
14. account_closure_time	(static)	octet-string			s.o.	x + 0x68
15. currency	(static)	structure			s.o.	x + 0x70
16. low_credit_threshold	(dyn.)	double-long			0	x + 0x78
17. next_credit_available_threshold	(dyn.)	double-long			0	x + 0x80
18. max_provision	(static)	long-unsigned			0	x + 0x88
19. max_provision_period	(static)	double-long			0	x + 0x90
Méthodes spécifiques		**o/f**				
1. activate_account (data)		f				x + 0x98
2. close_account (data)		f				x + 0xA0
3. reset_account (data)		f				x + 0xA8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2 Description d'attribut

4.6.2.2.1 logical_name

Identifie l'instance de l'objet "Account". Voir 6.2.17.

4.6.2.2.2 account_mode_and_status

Définit le payment_mode, énuméré ci-dessous, et l'état de "Account", également énuméré.

```
account_mode_and_status::= structure
{
payment_mode: enum:
                    (1) Mode de crédit,
                    (2) Mode de prépaiement
account_status: enum:
                    (1) Nouveau compte (inactif),
                    (2) Compte actif,
                    (3) Compte clos
}
```

Le payment_mode indique un prépaiement théorique ou un mode de crédit/paiement géré.

NOTE Payment_mode n'oblige pas à exécuter d'autres actions dans le compteur ni ne modifie son comportement.

Les objets "Credit" et "Charge" associés à un objet "Account" ne fonctionnent pas tant que l'objet "Account" n'est pas à l'état actif. Un nouvel objet "Account" (inactif) est passif et deviendra actif à *account_activation_time* ou à l'appel de la méthode *activate_account*.

4.6.2.2.3 current_credit_in_use

Cet attribut est un indice dans le *credit_reference_list* indiquant l'objet "Credit" en cours d'utilisation (*In use*).

4.6.2.2.4 current_credit_status

Cet attribut indique l'état de l'objet "Credit" actuel en cours d'utilisation et donne des informations relatives à l'objet "Credit" prioritaire suivant.

Bit 0 = in_credit, défini lorsque *available_credit* est supérieur à zéro,

Bit 1 = low_credit, défini lorsque le niveau d'*available_credit* de l'objet "Account" est inférieur à l'attribut *low_credit_threshold* de l'objet "Account",

NOTE 1 L'attribut *low_credit_theshold* de l'objet "Credit" associé *en cours d'utilisation* est repris dans cet attribut.

Bit 2 = next_credit_enabled, défini lorsque le *credit_reference_list* de l'objet "Account" contient au moins un autre objet "Credit" avec une priorité non nulle, quel que soit le niveau de crédit,

Bit 3 = next_credit_selectable, défini lorsque l'élément suivant de l'objet "Account" *credit_reference_list* contient un objet "Credit" de priorité inférieure avec une priorité non nulle et un bit 1 de *credit_configuration* (demande de confirmation) défini de telle sorte qu'il exige une confirmation (par exemple, un "crédit d'urgence" sélectionnable, mais qui n'a pas encore été sélectionné),

NOTE 2 L'objet "Credit" suivant devient sélectionnable lorsque l'objet "Credit" de priorité immédiatement supérieure a atteint le *next_credit_available_threshold* de l'objet "Account".

Bit 4 = next_credit_selected, défini lorsque le *credit_reference_list* de l'objet "Account" contient un objet "Credit" de priorité inférieure à l'objet "Credit" actuel en cours d'utilisation, et que l'objet "Credit" de priorité inférieure est à l'état *Selected/Invoked* (sélectionné/appelé)

NOTE 3 Il s'agit, par exemple, d'un "crédit d'urgence" qui a été sélectionné, mais qui n'est pas encore en cours d'utilisation (*In use*).

Bit 5 = selectable_credit_in_use, défini lorsque le crédit déjà sélectionné dans le *credit_reference_list* de l'objet "Account" est désormais en cours d'utilisation,

Bit 6 = out_of_credit, défini lorsque l'objet "Credit" qui était *en cours d'utilisation* est épuisé et qu'aucun autre crédit n'est disponible sans une action du consommateur ou d'un autre acteur.

Bit 7 = RÉSERVÉ

Si l'objet "Credit" de priorité suivante devient l'objet "Credit" en cours, tous les bits doivent être mis à jour.

NOTE 4 Un objet "Credit" devient sélectionnable (*Selectable*) lorsqu'il s'agit de l'objet "Credit" de priorité suivante et que l'attribut *next_credit_available_threshold* (qui reflète l'attribut *credit_available_threshold* de l'objet "Credit") de l'objet "Account" est supérieur ou égal à l'*available_credit* de l'objet "Account". Toutefois, si *available_credit* devient supérieur à *next_credit_available_threshold* à la suite de la sélection de "Credit", l'état du "Credit" reste (2) *Selected*. Si *available_credit* devient supérieur à *next_credit_available_threshold* à la suite d'un rechargement ou un appel de méthode pour augmenter le *current_credit_amount* d'un objet "Credit" qui est (2) *Selected* ou (3) *In use*, l'état de l'objet "Credit" devient (0) *Enabled.*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2.5 available_credit

L'attribut *available_credit* est égal à la somme des valeurs positives de *current_credit_amount* dans les instances de la classe "Credit" qui:

– figurent dans le *credit_reference_list* de l'objet "Account";
– et dont le *credit_status* d'un objet "Credit" est (2) *Selected/Invoked* ou (3) *In use*.

Cet attribut contient le montant global du crédit disponible associé à l'objet "Account". Cette valeur est positive ou nulle.

NOTE Les valeurs négatives des attributs *current_credit_amount* de l'objet "Credit" sont ajoutées dans *amount_to_clear*. Voir également l'axe vertical de la Figure 25.

double-long, mis à l'échelle en fonction de l'attribut *currency*.

4.6.2.2.6 amount_to_clear

Cette valeur est égale à la somme de:

– toutes les valeurs négatives de *current_credit_amount* dans les objets "Credit" qui:
  • figurent dans le *credit_reference_list* de l'objet "Account";
  • dont le *credit_status* de l'objet "Credit" est (4) *Exhausted* (Épuisé);
  • dont le bit 2 de *credit_configuration* (Exige le remboursement du montant du crédit) est effacé;
– la forme négative (Valeur * -1) du montant du crédit utilisé à partir de tous les objets "Credit" dont le bit 2 de *credit_configuration* (Exige le remboursement du montant du crédit) est défini;
– la forme négative (Valeur * -1) de la valeur de l'attribut *clearance_threshold* de l'objet "Account";

Voir aussi Figure 25.

NOTE 1 Les objets "Credit" dont le bit 2 de *credit_configuration* est défini sont appelés crédits "remboursables".

Le crédit utilisé est la différence entre le *preset_credit_amount* et le *current_credit_amount* (il s'agit donc d'une valeur positive) d'un objet "Credit" (3) *In use* (En cours d'utilisation) ou (4) *Exhausted* (Epuisé). Dans le cas des crédits non remboursables, le crédit utilisé n'est pas pertinent.

NOTE 2 Le processus d'application du compteur à paiement peut présenter un comportement fonctionnel particulier permettant au consommateur de contracter ou non un crédit d'urgence. Cette fonctionnalité n'entre pas dans le domaine d'application de la présente spécification d'accompagnement.

Cet attribut est important lorsque le compteur a accumulé une dette temporaire, par exemple, alors qu'un crédit d'urgence est en cours d'utilisation (*In use*) et/ou au cours d'une période de crédit à l'amiable.

Cet attribut contient le crédit minimal que le compteur doit recevoir (par réceptions de jeton et appels de méthode sur les objets "Credit") afin d'annuler les crédits négatifs et de réactiver les "Crédits" remboursables (un crédit d'urgence, par exemple) après qu'ils ont été à l'état (3) *In use* (En cours d'utilisation), (2) *Selected/Invoked* (Sélectionné/Appelé) ou (4) *Exhausted* (Épuisé).

NOTE 3 Pour obtenir une explication du processus de distribution des rechargements entre les objets "Credit", voir l'attribut 12 *token_gateway_configuration*.

NOTE 4 Si un compteur est utilisé en mode de prépaiement, *amount_to_clear* est en général le montant nécessaire à l'annulation d'une déconnexion. Le montant à effacer est présenté sur l'axe vertical de la Figure 25 sous forme de valeur négative.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

double-long, mis à l'échelle en fonction de l'attribut *currency* (4.6.2.2.15).

4.6.2.2.7 clearance_threshold

Cet attribut est utilisé conjointement avec *amount_to_clear*, et est inclus dans la description de cet attribut.

Cet attribut devient pertinent lorsqu'un compteur a consommé toutes les sources de crédit et qu'il contient des objets "Credit" dont les valeurs des attributs *current_credit_amount* sont inférieures à zéro.

Cela représente la valeur du crédit que l'attribut *available_credit* doit atteindre pour réactiver les "Crédits" remboursables.

Pour ce faire, il est évident qu'un crédit suffisant doit être ajouté au compteur afin de s'assurer que l'attribut *current_credit_available* de tous les objets "Credit" énumérés est supérieur ou égal à zéro.

double-long, mis à l'échelle en fonction de l'attribut *currency*.

4.6.2.2.8 aggregated_debt

Cet attribut est une simple somme des *total_amount_remaining* de tous les objets "Charge" qui figurent dans le *charge_reference_list* de l'objet "Account", le bit 1 (Collecte continue) de *charge_configuration* étant effacé.

NOTE Cette valeur n'est pas interchangeable avec *amount_to_clear*. Elle est fournie pour faciliter la comptabilité externe.

double-long, mis à l'échelle en fonction de l'attribut *currency*.

4.6.2.2.9 credit_reference_list

Cet attribut est un tableau de noms logiques, identifiant un ensemble d'objets "Credit" fonctionnant avec cet objet "Account". Les éléments du tableau doivent apparaître dans l'ordre de priorité (la priorité 1 étant la première priorité, n étant la dernière).

Les crédits de priorité 0 ne doivent pas figurer dans cette liste étant donné que, par définition, ils ne sont pas activés.

credit_reference_list::= array credit_reference

credit_reference::= octet-string

4.6.2.2.10 charge_reference_list

Cet attribut est un tableau de noms logiques, identifiant un ensemble d'objets "Charge" fonctionnant avec cet objet "Account". Les éléments du tableau doivent apparaître dans l'ordre de priorité (la priorité 1 étant la première priorité, n étant la dernière). Les charges de priorité 0 ne doivent pas figurer dans cette liste étant donné que, par définition, elles ne sont pas appliquées.

charge_reference_list::= array charge_reference

charge_reference::= octet-string

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2.11 credit_charge_configuration

Cet attribut indique les charges qui sont à collecter, et à partir de quels crédits.

Si le tableau ne contient pas d'élément, toutes les charges sont censées pouvoir être collectées à partir de tous les crédits, conformément au *credit_status (3) In use (En cours d'utilisation) ou (4) Exhausted (Épuisé)* des objets "Credit".

Si ce tableau contient des entrées, chacune d'elles représente chaque charge qui peut être collectée à partir de chaque crédit.

Un crédit à partir duquel aucune charge n'est collectée n'est pas consommé et ne sera pas modifié tant qu'une charge n'aura pas été configurée.

NOTE 1 La collecte d'une charge cesse lorsqu'elle devient nulle, lorsque le bit 1 (collecte continue) du *charge_configuration* de l'objet "Charge" approprié est effacé. Cela ne modifie pas le *credit_charge_configuration* de l'objet "Account".

credit_charge_configuration::= array credit_charge_configuration_element

credit_charge_configuration_element::= structure
{
credit_reference: octet-string,
charge_reference: octet-string,
collection_configuration: bit-string
}

où *credit_reference* et *charge_reference* contiennent le *logical_name* des objets "Credit" et "Charge" correspondants.

collection_configuration::= bit-string

Cet élément définit le comportement dans des conditions particulières.

Bit 0 = Collecter lorsque l'alimentation est déconnectée. S'il est défini, la "charge" indiquée dans *charge_reference* peut être collectée lorsque l'alimentation est déconnectée. S'il n'est pas défini, la "charge" indiquée dans *charge_reference* ne doit pas être collectée lorsque l'alimentation est déconnectée.

Bit 1 = Collecter pendant les périodes de limitation de la charge. S'il est défini, la "charge" indiquée dans *charge_reference* peut être collectée lorsque l'alimentation est dans une période de limitation de la charge. S'il n'est pas défini, la "charge" indiquée dans *charge_reference* ne doit pas être collectée lorsque l'alimentation est dans une période de limitation de la charge.

Bit 2 = Collecter pendant des périodes de crédit à l'amiable. S'il est défini, l'objet "Charge" indiqué dans *charge_reference* peut être collecté lorsque l'alimentation est dans une période de crédit à l'amiable. S'il n'est pas défini, la "charge" indiquée dans *charge_reference* ne doit pas être collectée lorsque l'alimentation est dans une période de crédit à l'amiable.

NOTE 2 L'application du compteur sait en interne s'il s'agit d'une alimentation déconnectée, d'une période de limitation de la charge ou d'une période de crédit à l'amiable, et entreprend l'action appropriée en fonction de la valeur de cet attribut.

NOTE 3 Les charges sont également implicitement collectées lorsque ces conditions spécifiques sont absentes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.2.2.12 token_gateway_configuration

Cet attribut permet de configurer la manière de ventiler un nouveau jeton de rechargement provenant d'un objet "Token gateway", de manière à répartir un pourcentage configurable du montant du jeton entre chaque objet "Credit".

NOTE 1 La répartition des montants du rechargement du jeton est également affectée par les valeurs des attributs *current_credit_amount*, *credit_type* et *credit_status* de l'objet "Credit".

Si la répartition du jeton de crédit reçu par l'intermédiaire de l'objet "Token gateway" fait l'objet de restrictions, cet attribut détermine la proportion minimale du jeton de crédit attribuée à chaque objet "Credit" référencé dans le tableau.

La somme des proportions indiquées dans ce tableau ne doit pas dépasser 100 %. Si la somme des proportions est inférieure à 100 %, le reste sera ajouté aux objets "Credit" à l'aide des règles ci-dessous pour un attribut *token_gateway_configuration* vide.

NOTE 2 Il est possible que certains "Credits" obtiennent plus que leur proportion allouée suite au processus présenté ci-dessus. Le jeton de crédit est toujours réparti à 100 % parmi les objets "Credit".

NOTE 3 La connexion entre un ou plusieurs objets "Token gateway" d'un compteur et un objet "Account" spécifique n'est pas explicitement présentée dans le modèle. La gestion de plusieurs objets "Account" et de plusieurs passerelles de jeton fait l'objet de spécifications d'accompagnement spécifiques au projet. Toutefois, normalement, un objet "Account" a un objet "Token gateway" et tous les jetons reçus suivent les règles associées à l'objet "Account" auquel il est associé (par l'application du champ du groupe de valeurs D; voir aussi 4.6.1).

```
token_gateway_configuration::= array
                               token_gateway_configuration_element

token_gateway_configuration_element::= structure
{
credit_reference          octet-string,
token_proportion          unsigned;
}
```

Où:

– credit_reference contient le *logical_name* de l'objet "Credit" et "Charge" correspondant;

– token_proportion est mis à l'échelle à un pour-cent par entier entre 0 et 100.

Si le tableau ne contient pas d'entrée, il est pris pour hypothèse que le traitement du jeton de crédit dans l'application de paiement du compteur ne fait l'objet d'aucune restriction, et il convient d'appliquer le crédit en premier lieu à l'objet "Credit" à priorité basse avec son *credit_status* défini sur (3) *In use* ou (4) *Exhausted*, puis ventilé vers les autres objets "Credit" dans l'ordre croissant de priorité de l'objet "Credit" (en commençant par la priorité n et en terminant par la priorité 1).

Si l'objet "Credit" exige un montant limité de rechargement (un objet "Crédit" d'urgence remboursé en totalité, par exemple), le crédit utilisé (*preset_credit_amount* moins *current_credit_amount* avant le rechargement) est prélevé du montant du jeton, et tous les excédents sont appliqués à des crédits à priorité élevée en suivant les mêmes règles. Lorsque le crédit utilisé est remboursé, "Credit" passe de (3) *In use* (En cours d'utilisation) ou (4) *Exhausted* (Epuisé) à (0) *Enabled* (Activé).

Voir aussi Note 1.

4.6.2.2.13 account_activation_time

Définit l'heure à laquelle l'objet lui-même appelle la méthode spécifique *activate_account*. Une définition avec une notation "not specified" (non spécifié) dans tous les champs de l'attribut ne sera pas examinée. Une notation "not specified" partielle, dans quelques champs de la date et de l'heure, n'est pas autorisée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si cette heure est passée, ce champ indique l'heure à laquelle l'objet "Account" a été activé.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date_time*.

4.6.2.2.14 account_closure_time

Définit l'heure à laquelle l'objet lui-même appelle la méthode spécifique *close_account*. Une définition avec une notation "not specified" (non spécifié) dans tous les champs de l'attribut ne sera pas examinée. Une notation "not specified" partielle, c'est-à-dire juste dans quelques champs de la date et de l'heure, n'est pas autorisée.

Si cette heure est passée, ce champ indique l'heure à laquelle l'objet "Account" a été fermé.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date_time*.

4.6.2.2.15 currency

Définit l'unité "currency" utilisée par toutes les fonctions d'un objet "Account". Le *charge_per_unit* de l'attribut *unit_charge* des objets "Charge" contient un facteur d'échelle supplémentaire qui peut être appliqué.

Les unités peuvent être une devise ou d'autres unités (kWh, minutes, par exemple) ou d'autres unités de consommation pour différents types de compteurs. Si les unités sont monétaires, alors le nom est exprimé par un symbole international à 3 caractères (GBP, USD, etc.) tel que défini dans l'ISO 4217.

```
currency::= structure
{
currency_name:      utf8-string,
currency_scale:     integer,
currency_unit:      enum
}
```

Où:

currency_name: utf8-string

selon l'ISO 4217

OU

min = minutes,

hrs = heures,

sec = secondes,

kWh = kilowatts-heures,

Whr = watts-heures,

mt3 = m^3,

ft3 = ft^3,

Jle = Joule

OU

Défini par la spécification d'accompagnement spécifique au projet, de telle sorte que tous les caractères utilisés soient en minuscules.

Le *currency_scale* indique l'exposant d'un multiple ou sous-multiple décimal de l'unité de base dans laquelle les valeurs d'attribut correspondantes sont exprimées. Les valeurs négatives indiquent les sous-multiples,

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE Par exemple: si la devise est l'Euro et que les valeurs sont exprimées en dixième de centime (un millième d'Euro), la valeur de l'échelle est -3 (moins 3).

currency_unit: enum:
(0) temps,
(1) consommation,
(2) monétaire

Le *currency_unit* est une valeur énumérée qui indique le type générique de la devise:

– temps (en minutes, secondes, heures, etc.);

– consommation (en kWh, Wh, m^3, Wh, etc.);

monétaire (GBP, USD, EUR, etc.).

4.6.2.2.16 low_credit_threshold

Cet attribut reflète l'attribut *warning_threshold* de l'objet "Credit" actuellement en cours d'utilisation. Ce seuil peut être utilisé pour générer un avertissement adressé au consommateur, par exemple. Il n'a pas d'autre fonction.

double-long, mis à l'échelle en fonction de l'attribut *currency*.

NOTE La valeur de cet attribut varie en fonction de l'objet "Credit" *en cours d'utilisation* au moment de la requête.

4.6.2.2.17 next_credit_available_threshold

Ce seuil reflète l'attribut credit_available_threshold de l'objet "Credit" de priorité suivante (sauf s'il n'y a pas d'objet "Credit" de ce type, auquel cas la valeur de cet attribut est -2 147 483 648 [valeur négative la plus grande possible]).

Lorsque l'attribut *available_credit* de l'objet "Account" devient inférieur ou égal à cette valeur, l'attribut *credit_status* de l'objet "Credit" de priorité suivante devient:

(1) *Selectable* (Sélectionnable) si le bit 1 (Demande confirmation) de l'attribut *credit_configuration* de l'objet "Credit" concerné est défini;

(2) *Selected/Invoked* (Sélectionné/Appelé) si le bit 1 (Demande de confirmation) est effacé et si next_credit_available_threshold est supérieur à zéro;

(3) *In use* (En cours d'utilisation) si le bit 1 (Demande de confirmation) est effacé et si next_credit_available_threshold est égal à zéro;

NOTE Le terme "priorité suivante" se rapporte au "crédit" suivant, dans l'ordre de priorité, qui n'a pas encore été sélectionné. Cet attribut varie en fonction du "crédit" qui vient ensuite dans l'ordre de priorité.

Le *next_credit_available_threshold* varie en fonction du "crédit" en cours d'utilisation.

double-long, mis à l'échelle en fonction de l'attribut *currency*.

4.6.2.2.18 max_provision

Cet attribut a un impact sur le fonctionnement des objets "Charge" configurés avec *charge_type* égal à (2) *payment_event_based_collection*.

Cet attribut contient le montant limite mis à l'échelle tel que défini dans l'attribut *currency* entre tous les objets "Charge" avec *charge_type* (2) *payment_event_based_collection*. La limite concerne la période définie dans *max_provision_period*.

NOTE 1 La combinaison de cet attribut avec *max_provision_period* a pour but de donner une limite à la valeur totale des collectes à partir des rechargements (en une semaine, par exemple).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE 2 Si un consommateur procède à plusieurs ventes sur une courte période, il est possible qu'il paye plus que ne l'exige le prestataire dans l'un de ces objets "Charge" instanciés (compte tenu du caractère imprévisible de la collecte en fonction du paiement).

long-unsigned, mis à l'échelle en fonction de l'attribut *currency*.

4.6.2.2.19 max_provision_period

Cet attribut contient la période applicable à l'attribut *max_provision*. Elle est spécifiée en secondes.

double-long

4.6.2.3 Description de la méthode

4.6.2.3.1 activate_account (data)

Cette méthode permet d'activer l'objet "Account". L'élément *account_status* de *account_mode_and_status* sera défini sur (2) *Account active* (Compte actif).

NOTE L'*account_activation_time* est défini sur l'heure d'appel de cette méthode.

data::= integer (0)

4.6.2.3.2 close_account (data)

Cette méthode force la clôture du compte.

NOTE 1 Cela peut être réalisé à la suite d'un changement de fournisseur, de locataire ou pour toute autre raison.

L'élément *account_status* de *account_mode_and_status* sera défini sur (3) *Account closed* (Compte clos). *account_closure_time* est défini sur l'heure d'appel de cette méthode.

NOTE 2 Si cette méthode est appelée, le compte n'est plus actif. Ses attributs cessent de changer avec les attributs de tous les objets "Credit" et "Charge" figurant dans *credit_reference_list* et *charge_reference_list* tant que l'objet "Account" n'est pas redevenu actif ou que les objets "Credit" et "Charge" ne sont pas référencés dans un autre objet "Account".

data::= integer (0)

4.6.2.3.3 reset_account (data)

Si account_status de l'attribut *account_mode_and_status* n'est pas égal à (2) *Active account* (Compte actif), cette méthode affecte les valeurs par défaut aux attributs de l'objet "Account" (sauf si un attribut n'a pas de valeur par défaut, auquel cas le comportement doit être spécifique au projet).

Si l'élément *account_status* de l'attribut *account_mode_and_status* est égal à 1 (Nouveau compte [inactif]), cette méthode ne doit avoir aucun effet.

data::= integer (0)

4.6.3 Credit interface class (class_id = 112, version = 0)

4.6.3.1 Généralités

Les instances de l'IC "Credit" permettent de gérer un crédit qui peut être consommé par des charges. Il existe plusieurs types de crédits différents, chaque objet "Credit" se caractérisant par les valeurs de ses attributs.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tous les "crédits" associés à une fourniture figurent dans l'attribut *credit_reference_list* de l'objet "Account". Les "crédits" passent d'un état à l'autre par:

- des rechargements;
- l'ajustement de current_credit_amount par appel de la méthode;
- la diminution du crédit par les charges.

Cela est expliqué dans le diagramme de transitions d'état en 4.6.3.2. Le paragraphe 4.6.1 répertorie les types de "crédit" tels que définis dans l'IEC TR 62055-21:2005.

4.6.3.2 États du crédit

Les états du crédit n'ont un sens que lorsque *priority* est non nul. Ils sont présentés dans le Tableau 34 et la Figure 24. Les transitions d'état sont présentées dans le Tableau 35.

Tableau 34 – États du crédit

Priorité	État du crédit	Signification
0	Any	L'instance de l'objet "Credit" est inactive. NOTE Si la priorité d'un "Crédit" est non nulle, mais que cela n'apparaît pas dans *credit_reference_list*, son comportement est le même que si sa priorité était nulle.
>0	(0) Enabled	La référence à l'objet "Credit" apparaît dans le *credit_reference_list* d'un objet "Account" actif avec une priorité non nulle.
>0	(1) Selectable	L'objet "Credit" exige certaines interactions supplémentaires avant de pouvoir passer à l'état *In use*. Un crédit est sélectionnable uniquement lorsqu'il s'agit du crédit de priorité suivante, s'il n'est pas épuisé et si le bit 1 (Demande de confirmation) de *credit_configuration* de l'objet "Credit" est défini.
>0	(2) Selected/Invoked	L'objet "Credit" qui était sélectionnable est à présent sélectionné, mais il peut ne pas encore être à l'état *In use* (un objet "Credit" de priorité plus élevée pouvant, par exemple, être encore utilisé, c'est-à-dire dans le cas d'EMC). Par ailleurs, un objet "Credit" qui était à l'état *Enabled* et qui n'exigeait aucune sélection peut arriver ici directement si le crédit de priorité supérieure est devenu *Exhausted*.
>0	(3) In use	L'objet "Credit" est en cours d'utilisation pour payer les *charges* dans le compteur.
>0	(4) Exhausted	L'objet "Credit" est épuisé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

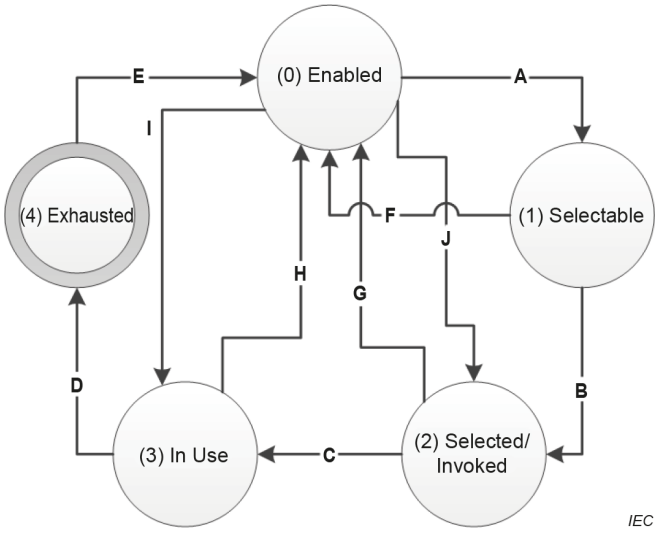


Figure 24 – États du Crédit lorsque la priorité > 0

Tableau 35 – Transitions d'état du crédit

État du crédit	De	À	Conditions
A	(0) Enabled	(1) Selectable	Un objet "Credit" devient sélectionnable s'il s'agit de l'objet "Credit" de priorité la plus élevée qui n'est pas épuisé et lorsque l'attribut *available_credit* de l'objet "Account" atteint l'attribut *next_credit_available_threshold* de l'objet "Account". NOTE 1 Cela s'applique uniquement si le bit 1 (Demande de confirmation) de l'attribut *credit_configuration* de l'objet "Credit" concerné est défini.
B	(1) Selectable	(2) Selected/ Invoked	Le déclencheur de transition est hors du domaine COSEM (un bouton-poussoir placé sur le compteur ou un autre moyen, par exemple). NOTE 2 Cela s'applique uniquement si le bit 1 (Demande de confirmation) de l'attribut *credit_configuration* de l'objet "Credit" concerné est défini.
C	(2) Selected/ Invoked	(3) In use	Cette transition se produit lorsque l'objet "Credit" de priorité précédente est épuisé. NOTE 3 À ce stade, l'attribut *next_credit_available_threshold* de l'objet "Account" est mis à jour pour tenir compte de l'attribut *credit_available_threshold* de l'objet "Credit" de priorité suivante.
D	(3) In use	(4) Exhausted	Cette transition se produit lorsque l'attribut *current_credit_amount* de l'objet "Credit" atteint le niveau défini dans l'attribut *limit*. NOTE 4 Cela peut indirectement provoquer la déconnexion de l'alimentation, en l'absence d'objet "Credit" de priorité inférieure passant à l'état *In use*. NOTE 5 Lorsque l'objet "Credit" en cours est épuisé, l'attribut *credit_status* est défini sur (4) *Exhausted*.
E	(4) Exhausted	(0) Enabled	Cette transition se produit lorsque l'attribut *current_credit_amount* est supérieur à l'attribut *limit* ou (dans le cas d'un crédit remboursable) que le crédit utilisé a été remboursé. NOTE 6 Cet objet "Credit" a reçu une recharge d'un certain montant provenant d'un jeton entrant ou d'un appel de méthode.
F	(1) Selectable	(0) Enabled	Cette transition se produit lorsque l'attribut *available_credit* de l'objet "Account" devient plus important que l'attribut *next_credit_available_threshold* de l'objet "Account". NOTE 7 Cela s'applique uniquement si le bit 1 (Demande de confirmation) de l'attribut *credit_configuration* de l'objet "Credit" concerné est défini. NOTE 8 C'est généralement le cas lorsque l'attribut *current_credit_amount* d'un objet "Credit" à l'état *In use* a été augmenté.
G	(2) Selected / Invoked	(0) Enabled	Cette transition se produit lorsque l'attribut *available_credit* de l'objet "Account" devient plus important que l'attribut *next_credit_available_threshold* de l'objet "Account". NOTE 9 C'est généralement le cas lorsque l'attribut *current_credit_amount* d'un objet "Credit" à l'état *In use* a été augmenté.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

État du crédit	De	À	Conditions
H	(3) In use	(0) Enabled	Cette transition se produit lorsque l'attribut *credit_status* d'un objet "Credit" de priorité supérieure passe à l'état *In use*. NOTE 10 C'est généralement le cas lorsque l'attribut *current_credit_amount* d'un objet "Credit" de priorité supérieure a été augmenté. A ce stade, l'attribut *next_credit_available_threshold* de l'objet "Account" change également.
I	(0) Enabled	(3) In use	Cette transition se produit lorsque l'objet "Credit" de priorité immédiatement supérieure est épuisé. NOTE 11 Cela s'applique uniquement lorsque le bit 1 (Demande de confirmation) de l'attribut *credit_configuration* de l'objet "Credit" concerné est effacé, que l'attribut *credit_available_threshold* de l'objet "Credit" est défini sur zéro et que son attribut *current_credit_amount* est supérieur à l'attribut *limit* (un crédit ne peut pas être à l'état *In use* tant que le crédit de priorité supérieure est à l'état *Exhausted*).
J	(1) Enabled	(2) Selected/ Invoked	La transition se produit lorsque l'attribut *available_credit* de l'objet "Account" devient inférieur à l'attribut *next_credit_available_threshold* de l'objet "Account". NOTE 12 Cela s'applique uniquement lorsque le bit 1 (Demande de confirmation avant de pouvoir passer à l'état Selected ou *In use*) de l'attribut *credit_configuration* de l'objet "Credit" concerné est effacé, et que l'attribut *credit_available_threshold* de cet objet "Credit" est supérieur à zéro.

4.6.3.3 Fanions d'état du crédit en cours

La signification des fanions *current_credit_status* (contenus dans l'attribut *current_credit_status* de l'objet "Account") est expliquée à la Figure 25.

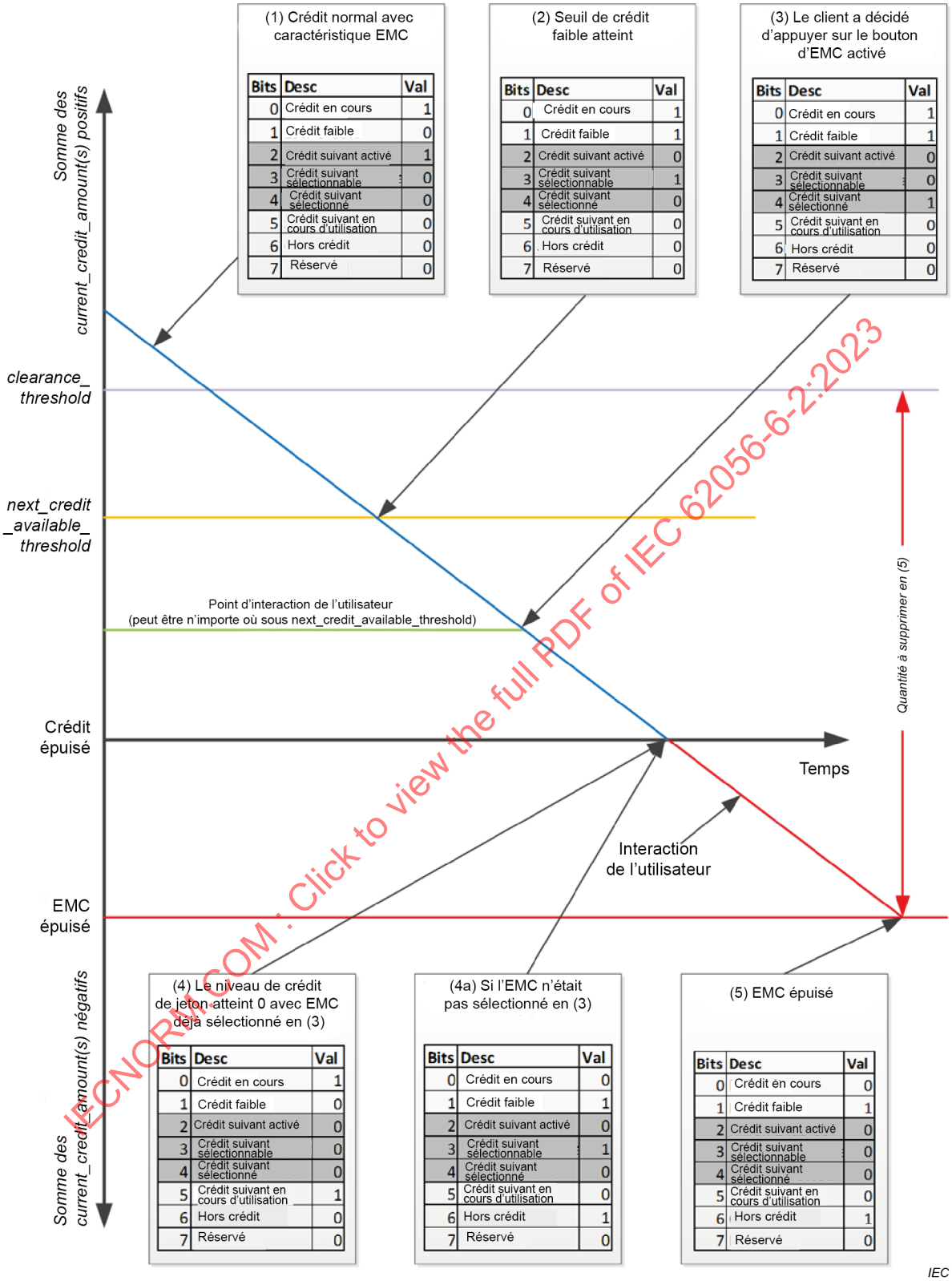
La Figure 25 donne un exemple de cas possibles concernant un système de comptage à paiement avec deux objets "Credit": Crédit de jeton et Crédit d'urgence.

NOTE Il s'agit d'une mise en œuvre possible. Le type et le nombre de crédits utilisés dans un projet réel font l'objet de spécifications d'accompagnement spécifiques au projet.

Deux options permettent de sélectionner des crédits sélectionnables: la sélection d'un objet "Credit" pendant que l'objet "Credit" en cours est à l'état *In use* (cas 4) ou lorsque l'objet "Credit" en cours est épuisé (cas 4a).

Dans le cas 4 (sélection d'un objet "Credit" pendant que l'objet "Credit" en cours est à l'état *In use*), le crédit commence à être consommé à partir du crédit suivant immédiatement après que le crédit en cours a été épuisé. Dans le cas 4a, le crédit en cours devient épuisé et continue potentiellement à être diminué par les charges jusqu'à ce que le consommateur choisisse le crédit sélectionnable ou ajoute une recharge.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023



Bits	Desc	Val
0	Crédit en cours	1
1	Crédit faible	0
2	Crédit suivant activé	1
3	Crédit suivant sélectionnable	0
4	Crédit suivant sélectionné	0
5	Crédit suivant en cours d'utilisation	0
6	Hors crédit	0
7	Réservé	0

Bits	Desc	Val
0	Crédit en cours	1
1	Crédit faible	1
2	Crédit suivant activé	0
3	Crédit suivant sélectionnable	1
4	Crédit suivant sélectionné	0
5	Crédit suivant en cours d'utilisation	0
6	Hors crédit	0
7	Réservé	0

Bits	Desc	Val
0	Crédit en cours	1
1	Crédit faible	1
2	Crédit suivant activé	0
3	Crédit suivant sélectionnable	0
4	Crédit suivant sélectionné	1
5	Crédit suivant en cours d'utilisation	0
6	Hors crédit	0
7	Réservé	0

Bits	Desc	Val
0	Crédit en cours	1
1	Crédit faible	0
2	Crédit suivant activé	0
3	Crédit suivant sélectionnable	0
4	Crédit suivant sélectionné	0
5	Crédit suivant en cours d'utilisation	1
6	Hors crédit	0
7	Réservé	0

Bits	Desc	Val
0	Crédit en cours	0
1	Crédit faible	1
2	Crédit suivant activé	0
3	Crédit suivant sélectionnable	1
4	Crédit suivant sélectionné	0
5	Crédit suivant en cours d'utilisation	0
6	Hors crédit	1
7	Réservé	0

Bits	Desc	Val
0	Crédit en cours	0
1	Crédit faible	1
2	Crédit suivant activé	0
3	Crédit suivant sélectionnable	0
4	Crédit suivant sélectionné	0
5	Crédit suivant en cours d'utilisation	0
6	Hors crédit	1
7	Réservé	0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 25 – Fonctionnement des fanions current_credit_status

Credit		0...n	class_id = 112, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. current_credit_amount	(dyn.)	double-long				x + 0x08
3. credit_type	(static)	enum				x + 0x10
4. priority	(static)	unsigned				x + 0x18
5. warning_threshold	(static)	double-long				x + 0x20
6. limit	(static)	double-long				x + 0x28
7. credit_configuration	(static)	bit-string				x + 0x30
8. credit_status	(dyn.)	enum				x + 0x38
L'attribut (9) suivant s'applique uniquement aux crédits d'urgence, à certains crédits périodiques (qui pourraient être des crédits réservés) et aux crédits à la consommation.						
9. preset_credit_amount	(static)	double-long				x + 0x40
10. credit_available_threshold	(static)	double-long				x + 0x48
L'attribut suivant s'applique uniquement aux crédits périodiques et aux crédits à la consommation.						
11. period	(static)	date-time				x + 0x50
Méthodes spécifiques		***o/f***				
1. update_amount (data)		f				x + 0x58
2. set_amount_to_value (data)		f				x + 0x60
La méthode suivante s'applique uniquement aux crédits d'urgence, aux crédits périodiques et aux crédits à la consommation.						
3. invoke_credit (data)		f				x + 0x68

4.6.3.4 Description d'attribut

4.6.3.4.1 logical_name

Identifie l'instance de l'objet "Credit". Voir 6.2.17.

4.6.3.4.2 current_credit_amount

Fournit la valeur de crédit de cet objet "Credit" particulier (voir la Figure 26).

NOTE Cette valeur est augmentée et diminuée en appelant les méthodes de cet objet, par des recharges et par la collecte des charges. Cette valeur participe à l'attribut *available_credit* de l'objet "Account" à partir duquel l'objet "Credit" est référence.

double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.4.3 credit_type

Il s'agit d'une énumération qui identifie le type de crédit que représente cet objet. Le type indique les attributs censés être traités pour cet objet "Credit", mais le traitement réel est fonction des valeurs des autres attributs contenant les fanions de configuration. Les types disponibles sont les suivants:

enum:

(0) token_credit,
(1) reserved_credit,
(2) emergency_credit,
(3) time_based_credit,
(4) consumption_based_credit

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.3.4.4 priority

Décrit la priorité d'activation de cet objet "Credit".

La valeur 1 est la priorité la plus élevée, la valeur 255 étant la plus basse.

Tous les objets "Credit" doivent avoir une priorité différente, sauf dans le cas de la priorité 0.

NOTE Le comportement est indéfini en présence de plusieurs objets "Credit" configurés avec la même priorité non nulle.

Une valeur 0 indique que l'objet "Credit" n'est jamais activé, même s'il peut toujours être configuré et que son attribut *current_credit_amount* peut être ajusté en appelant ses méthodes, mais il ne peut pas recevoir de rechargements.

Si un objet "Credit" de priorité zéro est configuré, il ne doit pas apparaître dans l'attribut *credit_reference_list* de l'objet "Account", ne sera pas pris en compte dans le calcul *available_credit* et ne doit pas être décrémenté par les charges.

Voir l'attribut *credit_reference_list* de l'objet "Account" (4.6.2.2.9) pour de plus amples informations.

4.6.3.4.5 warning_threshold

Contient une valeur de seuil pour *current_credit_amount*. Si *current_credit_amount* est réduit à la valeur de *warning_threshold*, un avertissement est adressé au consommateur lui indiquant que ce crédit est bas.

NOTE L'attribut *low_credit_theshold* de l'objet "Account" associé reprend ce même attribut de l'objet "Credit" à l'état *In use*.

double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.4.6 limit

Cet attribut contient une valeur de seuil pour *current_credit_amount*. Si *current_credit_amount* est réduit à la valeur de limit, *credit_status* passe à l'état (4) *Exhausted*.

NOTE Il est en général défini sur zéro dans un compteur fonctionnant en mode de prépaiement. Dans le cas d'un compteur fonctionnant en mode de crédit, la limite de l'objet "Credit" de priorité la plus élevée serait en principe égale au nombre négatif le plus important possible.

double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.4.7 credit_configuration

Permet de configurer le comportement de l'objet "Credit".

Si le **bit 0** est défini, la sélection de l'élément de crédit doit faire l'objet d'une indication visuelle sur l'écran du compteur.

Si le **bit 1** est défini, une confirmation est exigée de la part du consommateur pour accéder à ce type de "crédit" [c'est par exemple le cas d'un objet de crédit d'urgence, lorsque le consommateur doit appuyer sur un bouton avant que le "crédit" ne passe à l'état (2) Selected/Invoked].

Si le **bit 2** est défini, ceci indique que le montant de ce "crédit" doit être remboursé et, par conséquent, que le crédit utilisé va participer à l'attribut *amount_to_clear* de l'objet "Account" qui fait référence à l'objet "Credit" particulier.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si le **bit 3** est défini, l'attribut *current_credit_amount* peut être effacé par la fin de la période de facturation (à l'aide d'un script activant la méthode *set_amount_to_value*). Il n'est pas recommandé de configurer les "crédits" à cet effet, sauf si le compteur fonctionne en mode de crédit.

Si le **bit 4** est défini, cet objet "Credit" pourra recevoir un crédit à partir des jetons.

credit_configuration: bit-string:

Bit 0 = exige une indication visuelle;

Bit 1 = exige une confirmation avant de pouvoir être sélectionné/appelé; ou

Bit 2 = exige le remboursement du montant du crédit;

Bit 3 = réinitialisable;

Bit 4 = peut recevoir les montants du crédit à partir des jetons.

4.6.3.4.8 credit_status

Commandé par l'application de prépaiement pour indiquer l'état dans lequel se trouve l'objet "Credit".

Les états apparaissent sur la Figure 24 ci-dessus. Selon le type d'objet "Credit" et sa configuration, la valeur de cet attribut est commandée par l'application en fonction des valeurs des attributs *current_credit_amount, limit, preset_credit_amount* et *credit_available_threshold* des objets "Credit" et de l'attribut *amount_to_clear* de l'objet "Account".

Lorsque l'attribut *amount_to_clear* de l'objet "Account" qui référence les transitions de cet objet "Credit" à partir d'une valeur négative, l'état EMC doit de nouveau être (0) *Enabled*.

NOTE Lorsque l'attribut *credit_status* d'un objet "Credit" est à l'état (4) *Exhausted*, cet objet "Credit" ne peut pas être de nouveau appelé tant qu'il n'est pas passé à l'état (0) *Enabled*. S'il s'agit de l'objet "Credit" de priorité la plus basse, les charges peuvent toujours être appliquées à cet objet.

enum:

(0) l'instance de la classe de crédit est à l'état *Enabled*;

(1) l'instance de la classe de crédit est à l'état *Selectable*;

(2) l'instance de la classe de crédit est à l'état *Selected/Invoked*;

(3) l'instance de la classe de crédit est à l'état *In use* et peut être consommée par les charges;

(4) l'attribut *current_credit_amount* a été intégralement consommé et le crédit est considéré comme étant à l'état *Exhausted*.

Pour obtenir des explications supplémentaires, voir Tableau 34.

4.6.3.4.9 preset_credit_amount

Cet attribut est une valeur définie comme étant le montant initial du crédit disponible pour l'objet "Credit".

La valeur est ajoutée à l'attribut *current_credit_amount*:

a) lorsque *credit_status* passe à l'état (2) *Selected/Invoked* si le bit 1 (Demande de confirmation) de *credit_configuration* est défini, et en cas de confirmation de l'application; ou

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

b) lorsque *credit_status* passe à l'état (3) *In use* si le bit 1 (Demande de confirmation) de *credit_configuration* est effacé, sauf si *credit_status* est (4) *Exhausted*; ou

c) lorsque la méthode *invoke_credit* est appelée, si le bit 2 (Exige le remboursement du montant du crédit) de *credit_configuration* est défini; ou

d) lorsque le *date_time* de *period* se produit de manière implicite.

Si un "crédit" n'exige pas de montant prédéfini, l'attribut *preset_credit_amount* doit être 0 et l'objet "Credit" peut recevoir des montants de crédit provenant de jetons de crédit ou par appel des méthodes *update_amount* et *set_amount_to_value*. La valeur de *preset_credit_amount* ne change pas tant qu'une nouvelle valeur n'est pas écrite par le client.

Dans le cas des "crédits" du type *emergency_credit*:

– l'attribut *preset_credit_amount* doit être utilisé;

– le "crédit" doit pouvoir uniquement recevoir des montants de crédit provenant de jetons lorsque:

  - l'état est (3) *In use* ou (4) *Exhausted*;
  - tout ou partie du crédit a été utilisé;
  - s'il est à rembourser (le bit 2 [Exige le remboursement du montant du crédit] de *credit_configuration* est défini).

NOTE Si *current_credit_amount* a atteint *limit*, l'attribut *credit_status* passe à l'état (4) *Exhausted*. Si le bit 2 (Exige le remboursement du montant du crédit) de *credit_configuration* est défini, le crédit utilisé est égal à la différence entre *preset_credit_amount* et *current_credit_amount* (valeur positive), et il serait habituel de le rembourser en ajoutant un jeton de crédit ou en appelant une méthode. Suite au remboursement du "crédit", le *current_credit_amount* est nul, mais le *credit_status* sera (0) *Enabled* si *amount_to_clear* = 0 ou (4) *Exhausted* si *amount_to_clear* est inférieur à zéro.

Si cet attribut est défini sur zéro, il n'est associé à aucun comportement.

double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.4.10 credit_available_threshold

Valeur de seuil associée à *available_credit* de l'objet "Account".

Si *available_credit* de l'objet "Account" diminue jusqu'au *credit_available_threshold* de l'objet "Credit" de priorité suivante, le *credit_status* de cet objet devient:

a) 1 (Selectable) si le bit 1 (Demande de confirmation) de *credit_configuration* est défini; ou

b) 2 (Selected/Invoked) si le bit 1 (Demande de confirmation) de *credit_configuration* est effacé.

NOTE 1 Cela détermine si le "crédit" demande confirmation avant sa mise en service.

NOTE 2 Un objet "Credit" n'est pas à l'état *In use* tant qu'un "crédit" de priorité supérieure n'a pas été épuisé, mais s'il est sélectionné avant l'épuisement d'un "crédit" de priorité supérieure, la valeur de l'attribut *current_credit_amount* contribue au *available_credit* de l'objet "Account" associé.

double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.4.11 period

Si *credit_type* = 3 (time_based_credit) ou *credit_type* = 4 (consumption_based_credit), cet attribut contient l'heure à laquelle *current_credit_amount* est défini automatiquement sur *preset_credit_amount*.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date_time*. Les caractères génériques sont admis. Si tous les champs sont des caractères génériques, *preset_credit_amount* ne sera jamais ajouté.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.3.5 Description de la méthode

4.6.3.5.1 update_amount (data)

Cette méthode ajuste la valeur de l'attribut *current_credit_amount*. Les valeurs positives ajustent *current_credit_amount* positivement. Les valeurs négatives sont également admises.

data::= double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.5.2 set_amount_to_value (data)

Cette méthode définit la valeur de l'attribut *current_credit_amount*. Le montant indiqué dans l'attribut mis à jour doit être donné sous la forme d'un paramètre de réponse.

data::= double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

Retourne double-long, mis à l'échelle en fonction de l'attribut *currency* de l'objet "Account".

4.6.3.5.3 invoke_credit (data)

Cette méthode est appelée pour amener cet objet "Credit" sur *credit_status* (2) *Selected/Invoked* si son bit 1 (Demande de confirmation) de *credit_configuration* est défini et si *credit_status* est à l'état (1) *Selectable*.

Le mécanisme de sélection du "crédit" est hors du domaine d'application de COSEM et doit être spécifié par l'implémenteur (par exemple, bouton-poussoir, processus du compteur, script, etc.).

data::= integer (0)

4.6.3.6 Remarques supplémentaires

a) Même si les jetons sont souvent à l'origine de l'augmentation de *current_credit_amount* et si l'application des objets "Charge" provoque sa réduction, ces entrées sont appliquées au moyen d'un ajout signé, et il peut être possible d'accepter des jetons ou des objets "Charge" avec des valeurs négatives.

b) Le comportement du *type emergency_credit* est déterminé par l'attribut *preset_credit_amount* et par l'option "Requires the credit amount to be paid back" (Exige le remboursement du montant du crédit) de l'attribut *credit_configuration*.

   Si l'objet "Credit" est sélectionné/appelé, la valeur de l'attribut *preset_credit_amount* est ajoutée à la valeur de l'attribut *current_credit_amount* (qui est en principe nul à ce stade de son cycle de vie).

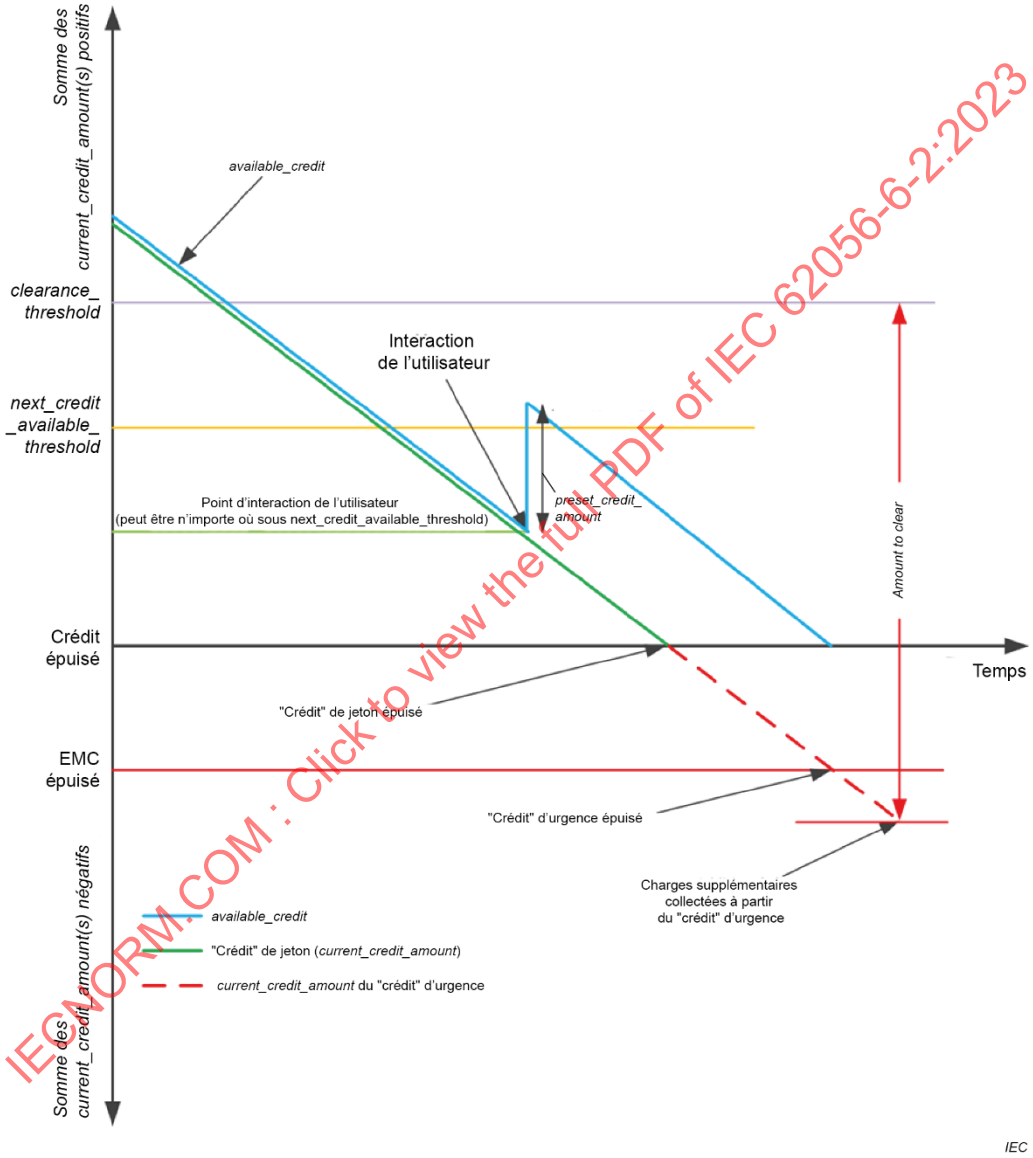
c) Pour reconstituer le crédit d'urgence et le faire passer à l'état (0) *Enabled* après avoir été à l'état (3) *In use*, il convient de recevoir un paiement suffisant pour amener l'*available_credit* de l'objet "Account" au moins jusqu'au *clearance_threshold*.

   Cela impliquera nécessairement d'apporter les fonds suffisants pour rembourser la valeur négative de *current_credit_amount* des objets "Credit" assurant la fonction EMC. Pour permettre la cogestion de plusieurs instances "Credit", les valeurs de crédit utilisées pour chaque objet "Credit" sont agrégées dans *amount_to_clear* avec l'attribut *clearance_threshold* de l'objet "Account" correspondant. Lorsque *amount_to_clear* devient nul, l'état (4) *Exhausted* est par définition effacé sur tous les objets "Credit" associés.

d) L'attribut *token_gateway_configuration* de l'objet "Account" détermine la distribution du crédit aux objets "Credit".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

La Figure 26 présente l'interaction des attributs *current_credit_amount* de deux objets "Credit" dans un système de comptage à paiement. Elle montre que le crédit d'urgence, lorsqu'il est sélectionné, contribue à *available_credit*, mais ne passe pas à l'état *In use* tant que le "crédit" de jeton n'est pas à l'état *Exhausted*. Le "crédit" d'urgence commence à contribuer à *amount_to_clear* lorsqu'il passe à l'état *In use*. Lorsque le "crédit" d'urgence est à l'état *In use* et qu'il contribue à *amount_to_clear*, ce dernier tient également compte du seuil de suppression. Le seuil de suppression est uniquement important lorsque *current_credit_amount* du "crédit" de jeton est inférieur ou égal à zéro.



IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Figure 26 – Interaction de current_credit_amount et d'available_credit avec le "crédit" de jeton et le "crédit" d'urgence

A la suite de l'appel du "crédit" d'urgence, *available_credit* devient plus important que *next_credit_available_threshold*, mais le "crédit" d'urgence ne fait pas passer credit_status de (1) *Selected* à (0) *Enabled*. Toutefois, en cas de rechargement ou d'appel de la méthode sur *set_amount_to_value*, et si *available_credit* est forcé de dépasser *next_credit_available_threshold* (en augmentant le *current_credit_amount* d'un objet "Credit"), l'état doit être forcé à (0) *Enabled*.

4.6.4 Charge (class_id = 113, version = 0)

4.6.4.1 Généralités

Les instances de l'IC "Charge" permettent de gérer une seule charge. En fonction des attributs configurés (le montant par prix et la période, par exemple), la charge est prélevée au moment opportun de l'objet "Credit" à l'état *In use*.

NOTE Les détails du cycle de collecte (prélèvement de charge) peuvent dépendre du projet et font donc l'objet de spécifications d'accompagnement spécifiques au projet étant donné qu'ils ont une influence sur les estimations des valeurs en cours par les tiers.

Chaque objet "Charge" se caractérise par les valeurs de ses attributs.

Toutes les "charges" associées à une fourniture sont référencées dans l'attribut *charge_reference_list* de l'objet "Account" correspondant (4.6.2.2.10).

Charge		**0...n**	**class_id = 113, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. total_amount_paid	(dyn.)	double-long				x + 0x08
3. charge_type	(static)	enum				x + 0x10
4. priority	(static)	unsigned				x + 0x18
5. unit_charge_active	(static)	structure				x + 0x20
6. unit_charge_passive	(static)	structure				x + 0x28
7. unit_charge_activation_time	(static)	octet-string				x + 0x30
L'attribut suivant fait uniquement référence aux collectes périodiques et aux collectes basées sur la consommation.						
8. period	(static)	double-long-unsigned				x + 0x38
Les attributs suivants concernent tous les types de charges.						
9. charge_configuration	(static)	bit-string				x + 0x40
10. last_collection_time	(dyn.)	date-time				x + 0x48
11. last_collection_amount	(dyn.)	double-long				x + 0x50
12. total_amount_remaining	(dyn.)	double-long				x + 0x58
Les attributs suivants concernent uniquement la collecte en fonction des événements de paiement.						
13. proportion	(static)	long-unsigned				x + 0x60
Méthodes spécifiques		***o/f***				
1. update_unit_charge (data)		f				x + 0x68
2. activate_passive_unit_charge (data)		f				x + 0x70
Les méthodes suivantes concernent uniquement la collecte périodique et la collecte en fonction des événements de paiement.						
3. collect (data)		f				x + 0x78
Les méthodes suivantes concernent uniquement la collecte en fonction des événements de paiement.						
4. update_total_amount_remaining (data)		f				x + 0x80
5. set_total_amount_remaining (data)		f				x + 0x88

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.4.2 Description d'attribut

4.6.4.2.1 logical_name

Identifie l'instance de l'objet "Charge". Voir 6.2.17.

4.6.4.2.2 total_amount_paid

Contient le montant total collecté pour cet objet "Charge". En principe, il n'est pas réinitialisé lorsque l'objet "Account" reste actif.

NOTE Cette valeur est incrémentée par l'application au même moment et au même taux que la charge collectée.

4.6.4.2.3 charge_type

Indique le type de collecte associée à cette instance de la classe "Charge".

enum:

(0) consumption_based_collection;
(1) time_based_collection;
(2) payment_event_based_collection.

charge_type indique les attributs dont le traitement est prévu pour cet objet "Charge". Le traitement est également influencé par l'attribut *charge_configuration*.

NOTE Dans le cas des charges *payment_event_based_collection*, cet attribut dicte le mécanisme de collecte de charges. L'application collecte les charges *payment_event_based_collection* des "crédits" appropriés dès qu'un crédit est distribué à partir d'un nouveau jeton.

4.6.4.2.4 priority

Décrit la priorité de cette "charge" particulière lors de la collecte à partir d'un objet "Credit".

Tous les objets "Charge" doivent avoir une priorité différente, sauf dans le cas de la priorité 0.

La valeur 1 représente la priorité la plus élevée, la valeur 255 étant la plus basse.

Une valeur 0 indique que l'objet "Charge" n'est pas activé, même s'il peut toujours être configuré et que son attribut *total_amount_remaining* peut être ajusté en appelant les méthodes appropriées de l'objet "Charge".

Les objets "Charge" avec la valeur de priorité 0 ne doivent pas apparaître dans le *charge_reference_list* de l'objet "Account", mais les objets "Charge" de priorité <> 0 n'apparaissent pas nécessairement dans le *charge_reference_list* d'un objet "Account".

4.6.4.2.5 unit_charge_active

Définit le prix actif, qui est le montant imputé par unité consommée, par unité de temps ou par paiement reçu, en termes d'attribut *currency* de l'instance "Account" pertinente et, le cas échéant, d'attribut *scaler_unit* de l'objet identifié par la structure *commodity_reference*.

```
unit_charge_active::= structure
{
charge_per_unit_scaling:    charge_per_unit_scaling_type,
commodity_reference:        commodity_reference_type,
charge_table:               charge_table_type
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Où:

```
charge_per_unit_scaling_type::= structure
{
        commodity_scale :       integer,
        price_scale:            integer
}
commodity_reference_type::=     structure
{
        class_id:               long-unsigned,
        logical_name:           octet-string,
        attribute_index:        integer
}

charge_table_type::= array      charge_table_element

charge_table_element::=         structure
{
        index:                  octet-string,
        charge_per_unit:        long
}
```

a) Explications relatives à charge_per_unit_scaling

Si charge_type = (0) *consumption_based_collection*, le champ de *charge_per_unit_scaling* est exprimé sous la forme d'un exposant (base 10) du facteur de multiplication de l'unité de base dans laquelle les valeurs d'attribut pertinentes sont exprimées en interne:

– commodity_scale est appliqué aux unités associées à *commodity_reference*;

– price_scale est appliqué à *currency_scale* de l'attribut *currency* de l'objet "Account" correspondant.

Si *charge_type* N'EST PAS (0) *consumption_based_collection*, *commodity_scale* doit être défini sur 0 et *price_scale* est exprimé sous la forme d'un exposant (base 10) de *currency_scale* à appliquer à l'attribut *currency* de l'objet "Account".

NOTE 1 Par exemple, soit un *currency_name* primaire exprimé en Euros (comme indiqué dans l'objet "Account"), contenant des valeurs en unités d'un centime (un centième d'Euro), et soit une marchandise fournie sous forme d'énergie électrique d'importation active dont les valeurs exprimées en unités de 1 000 Wh (un kilowatt-heure) depuis l'attribut *scaler_unit* de *commodity_reference*. Ensuite, pour un prix en dixièmes de centime par kilowatt-heure, l'échelle de prix comportera une valeur –3 (moins trois) étant donné qu'il s'agit de millièmes d'une unité de base, l'échelle de marchandise ayant une valeur 0 (zéro) étant donné qu'elle est en unités de base (c'est-à-dire kWh).

Si le "currency" primaire est kWh ou équivalent par rapport à *commodity_scale*, un tarif TOU n'est pas possible, et l'élément *charge_table* d'*unit_charge_active* et d'*unit_charge_passive* ne doit pas être pertinent, mais *commodity_scale* et *price_scale* doivent avoir la même valeur.

NOTE 2 *commodity_scale* et *price_scale* n'impliquent pas de taux de collecte particulier.

b) Explications relatives à *charge_type*

Si *charge_type* = (0) *consumption_based_collection*, *commodity_reference* identifie un attribut *scaler_unit* d'un registre total, en mesurant tout ce qui est fourni (l'énergie électrique importée, par exemple).

Si *charge_type* = (1) *time_based_collection* ou (2) *payment_event_based_collection*, *commodity_reference* doit être une structure de zéro élément.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

c) Explications des éléments charge_table

Chaque élément du tableau *charge_table* est une structure qui identifie ce qui fait l'objet de charges et d'une collecte.

Si *charge_type* = (0) *consumption_based_collection*:

– l'indice est un identifiant d'un dérivé de la principale référence de marchandise (les registres de tarification, par exemple);
– *charge_per_unit* est le montant de charge par unité qui est mis à l'échelle selon les facteurs *charge_per_unit_scaling*.

NOTE 3 Par exemple, si le consommateur final génère de l'énergie, il convient que "Charge" soit négatif, ce qui signifie que l'entreprise de gestion d'électricité paye le consommateur final pour l'énergie qu'il a produite.

NOTE 4 Il est également possible de modéliser l'énergie exportée à l'aide d'un objet "Account" distinct et d'autres objets connexes.

Si *charge_type* = (1) *time_based_collection* ou (2) *payment_event_based_collection*, une seule entrée dans le tableau *charge_table* est créée, contenant:

– l'indice est un octet-string de longueur 0;
– *charge_per_unit* est une valeur équivalente au montant imputé par *period* ou par événement de paiement et mise à l'échelle selon les facteurs *charge_per_unit_scaling*.

4.6.4.2.6 unit_charge_passive

Contient les détails des prix à appliquer dès l'occurrence de la date d'activation.

Sa structure de données est la même que *unit_charge_active*. Sur activation, l'ensemble de la structure d'*unit_charge_passive* est copié dans *unit_charge_active*.

4.6.4.2.7 unit_charge_activation_time

Définit l'heure à laquelle l'objet lui-même appelle la méthode spécifique *activate_unit_charge_passive*.

Une définition avec une notation "not specified" (non spécifié) dans tous les champs de l'attribut désactivera cet automatisme. Une notation "not specified" partielle, c'est-à-dire juste dans quelques champs de la date et de l'heure, n'est pas autorisée.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date_time*.

4.6.4.2.8 period

Si *charge_type* = (0) *consumption_based_collection* ou (1) *time_based_collection*, il définit la période pendant laquelle la charge sera collectée. Si la période est définie sur zéro, cette "charge" ne fait l'objet d'aucune collecte automatique: la collecte est assurée en appelant la méthode *collect*.

NOTE Les mises en œuvre peuvent varier: il est possible de collecter l'ensemble de la charge en une seule collecte ou de la répartir sur plusieurs parties de collecte.

Si *charge_type* = (2) *payment_event_based_collection*, cet attribut n'est pas utilisé.

double-long-unsigned, en secondes.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.4.2.9 charge_configuration

Cet attribut est une chaîne binaire, comme suit:

charge_configuration::= bit-string

Bit 0 = collecte basée sur le pourcentage;

Bit 1 = collecte continue.

Si le bit 0 est défini et que *charge_type* = (2) *payment_event_based_collection*, cet objet "Charge" collectera une proportion de chaque rechargement conformément à l'attribut *proportion* (4.6.4.2.13). Cela s'applique uniquement aux rechargements, et non aux appels de méthode.

NOTE L'attribut proportion de l'objet "Charge" et l'attribut provision de l'objet "Account" donnent des informations relatives à cette collecte.

Si le bit 1 est défini, la collecte ne se termine pas lorsque l'attribut *total_amount_remaining* est égal à zéro. Si le bit 1 est effacé, la collecte de charge se termine lorsque l'attribut *total_amount_remaining* est égal à zéro.

4.6.4.2.10 last_collection_time

Contient le *date_time* indiquant l'instant où la dernière collecte a eu lieu.

Cela inclut la collecte incrémentielle en fonction de la consommation.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date_time*.

4.6.4.2.11 last_collection_amount

Contient le montant de la dernière collecte lié à l'attribut *last_collection_time*.

double-long, mis à l'échelle conformément à l'élément *price_scale* d'*unit_charge_active*.

4.6.4.2.12 total_amount_remaining

Si le bit 1 (Collecte continue) de *charge_configuration* est effacé, cet attribut contient le montant total restant pour cet objet "Charge".

NOTE 1 La valeur de cet attribut est à l'origine configurée en appelant la méthode *set_total_amount_remaining*.

NOTE 2 Cet attribut est utilisé pour limiter les collectes au remboursement d'une dette. Il ne s'agit pas d'une mesure directe de la responsabilité du consommateur. La valeur ne peut pas être négative (elle est plutôt nulle) et n'est pas incrémentée par la collecte d'une "charge" avec un prix négatif.

Si le bit 1 (Collecte continue) de *charge_configuration* est défini, cet attribut est nul.

NOTE 3 Voir aussi les notes supplémentaires à la fin de la spécification de l'IC "Charge".

double-long, mis à l'échelle conformément à l'élément *price_scale* d'*unit_charge_active*.

4.6.4.2.13 proportion

Si le bit 0 (Collecte basée sur le pourcentage) de *charge_configuration* est défini et que *charge_type* = (2) *payment_event_based_collection*, cet attribut est la proportion de chaque montant de rechargement traité dans l'objet "Token gateway" lié à cet objet "Charge" par l'intermédiaire de l'objet "Account".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

La plage 0x0000 à 0x2710 (0 à 10 000) représente 0 à 100 %.

NOTE Le montant de la collecte qui a lieu peut être limité par les attributs *max_provision* et *max_provision_period* de l'objet "Account".

S'il est exigé de collecter un montant fixe à chaque rechargement (*charge_type* = 2), il convient que le montant à prélever soit défini dans le premier élément du tableau *charge_table* de l'attribut *unit_charge_active*/*unit_charge_passive*.

Si *charge_type* <> (2) *payment_event_based_collection* ou que le bit 0 (Collecte basée sur le pourcentage) de *charge_configuration* est effacé, cet attribut n'a aucun effet.

Si *charge_type* = (2) *payment_event_based_collection* et que la collecte d'un montant fixe est exigée, voir *unit_charge_active* (4.6.4.2.5) et *unit_charge_passive* (4.6.4.2.6).

4.6.4.3 Description de la méthode

4.6.4.3.1 update_unit_charge (data)

Permet de mettre à jour un sous-ensemble de valeurs de charge unitaire à l'intérieur de l'attribut *unit_charge_passive*. Les valeurs de *charge_per_unit_scaling* et de *commodity_reference* ne sont pas affectées par cette méthode.

Il reste nécessaire d'activer l'*unit_charge_passive* afin que les valeurs puissent prendre effet.

```
data::= array charge_table_element
charge_table_element::= structure
{
index:                  octet-string,
charge_per_unit:        long
}
```

Voir *charge_table_element* de l'attribut *unit_charge_active* pour une description des éléments.

NOTE Si l'indice existe, la valeur est écrasée. S'il n'existe pas, la nouvelle valeur est ajoutée au tableau.

4.6.4.3.2 activate_passive_unit_charge (data)

Copie l'ensemble de la structure de l'attribut *unit_charge_passive* dans l'attribut *unit_charge_active*.

NOTE Cette méthode n'a aucun impact sur l'activité de collecte ou sur la priorité de l'objet "Charge".

```
data::= integer (0)
```

4.6.4.3.3 collect (data)

Si *charge_type* <> (0) *consumption_based_collection*, cette méthode collecte le montant défini dans *unit_charge_active* lorsque le bit 0 (Collecte basée sur le pourcentage) de *charge_configuration* est effacé.

Si *charge_type* = (0) *consumption_based_collection*, cette méthode n'a aucun effet.

```
data::= integer (0).
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.4.3.4 update_total_amount_remaining (data)

Permet de mettre à jour l'attribut *total_amount_remaining*. La valeur est à mettre à l'échelle conformément à *price_scale* d'*unit_charge_active*. La valeur est ajoutée à *total_amount_remaining*. Le montant indiqué dans l'attribut *total_amount_remaining* doit être donné sous la forme d'un paramètre de réponse.

NOTE Cela n'a aucun impact sur *total_amount_paid*.

data::= double-long, mis à l'échelle conformément à *price_scale* d'*unit_charge_active*.

et

retourne double-long, mis à l'échelle conformément à *price_scale* d'*unit_charge_active*.

4.6.4.3.5 set_total_amount_remaining (data)

Définit l'attribut *total_amount_remaining*. La valeur ne doit pas être inférieure à 0. Le montant indiqué dans l'attribut *total_amount_remaining* doit être donné sous la forme d'un paramètre de réponse.

NOTE *total_amount_remaining* est défini sans affecter *total_amount_paid*.

data::= double-long, mis à l'échelle conformément à *price_scale* d'*unit_charge_active*.

et

retourne double-long, mis à l'échelle conformément à *price_scale* d'*unit_charge_active*.

4.6.5 Token gateway (class_id = 115, version = 0)

4.6.5.1 Généralités

Une instance de l'IC "Token gateway" met en œuvre l'interface de support de jeton.

NOTE Une seule instance de l'objet "Token gateway" est instanciée pour chaque objet "Account" et, par conséquent, chaque contrat de fourniture.

Token gateway		**0...n**	**class_id = 115, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. token	(dyn.)	octet-string				x + 0x08
3. token_time	(dyn.)	octet-string				x + 0x10
4. token_description	(dyn.)	array				x + 0x18
5. token_delivery_method	(dyn.)	enum				x + 0x20
6. token_status	(dyn.)	structure				x + 0x28
Méthodes spécifiques		***o/f***				
1. enter (data)		**o**				x + 0x30

4.6.5.2 Description d'attribut

4.6.5.2.1 logical_name

Identifie l'instance de l'objet "Token gateway". Voir 6.2.17.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.6.5.2.2 token

Contient la chaîne d'octets non traitée du dernier jeton reçu pour les besoins de la capture dans le profil d'historique.

4.6.5.2.3 token_time

Contient, pour le dernier jeton reçu et traité, l'heure et la date auxquelles un jeton reçu est arrivé au niveau du serveur DLMS®/COSEM.

octet-string, mis en forme comme indiqué en 4.1.6.1 pour *date_time*.

4.6.5.2.4 token_description

Contient une description du dernier jeton reçu et traité fourni par le programme d'application du compteur.

NOTE Par exemple, il peut contenir le type de jeton (crédit ou ingénierie) et/ou l'origine du jeton.

L'utilisation de cet attribut doit être décrite dans la spécification du jeton ou dans les spécifications d'accompagnement spécifiques au projet.

token_description:: = array token_description_element

token_description_element::= octet-string.

4.6.5.2.5 token_delivery_method

Reflète le chemin de réception du dernier jeton.

enum:

(0) par communications distantes;
(1) par communications locales;
(2) par entrée manuelle.

4.6.5.2.6 token_status

token_status est une structure contenant une énumération générique qui indique l'état de fonctionnement du dernier jeton et une chaîne binaire facultative qui peut être associée à l'état du jeton, donnant des informations supplémentaires relatives au *status_code* du jeton.

NOTE 1 Il est prévu que le contenu de la chaîne binaire facultative soit documenté dans une spécification d'accompagnement spécifique au projet.

NOTE 2 La signification exacte et les critères d'évaluation des valeurs de *status_code* sont sensiblement différents selon le protocole et le format du jeton. Par exemple, un jeton conforme à l'IEC 62055-41 comporte des critères et des significations très précis pour chacun des termes de résultat de validation, d'authentification et de jeton, alors que d'autres spécifications peuvent avoir des termes différents.

```
token_status::= structure
{
        status_code:            enum,
        data_value:             bit-string
}
```

status_code: enum:

(0) résultat du format de jeton OK;
(1) résultat de l'authentification OK;
(2) résultat de la validation OK;
(3) résultat de l'exécution du jeton OK;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(4) échec du format de jeton;
(5) échec de l'authentification;
(6) échec du résultat de la validation;
(7) échec du résultat de l'exécution du jeton;
(8) jeton reçu, mais pas encore traité.

A la réception d'un jeton, l'état initial doit être positionné à 8, alors que le jeton est en cours de traitement et jusqu'à ce qu'un autre état puisse être déduit.

L'utilisation du champ de chaîne binaire data_value doit être définie dans les spécifications supplémentaires d'accompagnement au projet.

4.6.5.3 Description de la méthode

4.6.5.3.1 enter (data)

Cette méthode est appelée pour transférer un jeton vers le serveur DLMS®/COSEM sous la forme d'une chaîne d'octets. En cas de succès, elle peut renvoyer l'attribut *token_status*.

data::= octet-string, et retourne *token_status*

4.6.5.4 Remarques supplémentaires

L'objet "Data" peut contenir un certain nombre de limites configurables, relatives au comportement et au traitement des jetons et à certains aspects du système de comptage à paiement. Certaines de ces limites ont été définies dans la présente spécification, ont des codes OBIS et sont décrites ci-dessous:

Max credit limit: À la réception d'un jeton, le processus d'application du compteur doit vérifier que l'ajout de la valeur du jeton de crédit aux objets de crédit associés ne force pas *available_credit* de l'objet "Account" à dépasser la limite programmée dans l'objet "Max credit limit". Si le jeton reçu semble forcer *available_credit* à dépasser cette limite, le compteur doit le rejeter et renvoyer l'état approprié.

Max vend limit: À la réception d'un jeton, le processus d'application du compteur doit vérifier que la valeur du jeton de crédit ne dépasse pas la limite programmée dans l'objet "Max vend limit". Si le jeton reçu semble dépasser cette limite, le compteur doit le rejeter et renvoyer l'état approprié.

Voir aussi 6.2.17.

4.7 Classes d'interfaces pour l'établissement d'échange de données via les ports locaux et les modems

4.7.1 IEC local port setup (class_id = 19, version = 1)

4.7.1.1 Vue d'ensemble

Cette IC permet de modéliser la configuration de ports de communication en utilisant les protocoles spécifiés dans l'IEC 62056-21:2002. Plusieurs ports peuvent être configurés.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC local port setup		0...n	class_id = 19, version = 1			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. default_mode	(static)	enum				x + 0x08
3. default_baud	(static)	enum				x + 0x10
4. prop_baud	(static)	enum				x + 0x18
5. response_time	(static)	enum				x + 0x20
6. device_addr	(static)	octet-string				x + 0x28
7. pass_p1	(static)	octet-string				x + 0x30
8. pass_p2	(static)	octet-string				x + 0x38
9. pass_w5	(static)	octet-string				x + 0x40
Méthodes spécifiques		***o/f***				

4.7.1.2 Description d'attribut

4.7.1.2.1 logical_name

Identifie l'instance d'objet "IEC local port setup". Voir 6.2.18.

4.7.1.2.2 default_mode

Définit le protocole utilisé par le compteur sur le port.

enum:

(0) protocole conforme à l'IEC 62056-21:2002 (modes A…E);
(1) protocole conforme à l'IEC 62056-46:2002/AMD1:2006. En utilisant cette valeur d'énumération, tous les autres attributs de cette IC ne sont pas applicables;
(2) protocole non spécifié. En utilisant cette valeur d'énumération, l'attribut 4), prop_baud, est utilisé pour régler la vitesse de communication sur le port. Tous les autres attributs ne sont pas applicables.

4.7.1.2.3 default_baud

Définit le débit en bauds pour la séquence d'ouverture.

enum:

(0) 300 bauds;
(1) 600 bauds;
(2) 1 200 bauds;
(3) 2 400 bauds;
(4) 4 800 bauds;
(5) 9 600 bauds;
(6) 19 200 bauds;
(7) 38 400 bauds;
(8) 57 600 bauds;
(9) 115 200 bauds.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.1.2.4 prop_baud

Définit le débit en bauds à proposer par le compteur.

enum:

(0) 300 bauds;
(1) 600 bauds;
(2) 1 200 bauds;
(3) 2 400 bauds;
(4) 4 800 bauds;
(5) 9 600 bauds;
(6) 19 200 bauds;
(7) 38 400 bauds;
(8) 57 600 bauds;
(9) 115 200 bauds.

4.7.1.2.5 response_time

Définit le temps minimal entre la réception d'une requête (fin de télégramme de requête) et l'émission de la réponse (début de télégramme de réponse).

enum:

(0) 20 ms;
(1) 200 ms.

4.7.1.2.6 device_addr

Adresse de dispositif selon l'IEC 62056-21:2002.

4.7.1.2.7 pass_p1

Mot de passe 1 selon l'IEC 62056-21:2002.

4.7.1.2.8 pass_p2

Mot de passe 2 selon l'IEC 62056-21:2002.

4.7.1.2.9 pass_w5

Mot de passe W5 réservé pour des applications nationales.

4.7.2 IEC HDLC setup (class_id = 23, version = 1)

4.7.2.1 Vue d'ensemble

Cette IC permet de modéliser et de configurer des voies de communication selon l'IEC 62056-46:2002/A1:2006. Plusieurs voies de communication peuvent être configurées.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC HDLC setup		0...n	class_id = 23, version = 1			
Attributs		***Type de données***	**Min.**	**Max.**	**Def.**	**Nom court**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	9	5	x + 0x08
3. window_size_transmit	(static)	unsigned	1	7	1	x + 0x10
4. window_size_receive	(static)	unsigned	1	7	1	x + 0x18
5. max_info_field_length_transmit	(static)	long-unsigned	32	2030	128	x + 0x20
6. max_info_field_length_receive	(static)	long-unsigned	32	2030	128	x + 0x28
7. inter_octet_time_out	(static)	long-unsigned	20	6000	25	x + 0x30
8. inactivity_time_out	(static)	long-unsigned	0		120	x + 0x38
9. device_address	(static)	long-unsigned	0x0010	0x3FFD		x + 0x40
Méthodes spécifiques		***o/f***				

NOTE 1 La valeur maximale des attributs *max_info_field_length_transmit* et *max_info_field_length_receive* a été augmentée de 128 à 2 030 pour des raisons d'efficacité.

NOTE 2 Un *max_info_field_length_receive* de 128 octets est nécessaire afin d'assurer une performance minimale.

NOTE 3 La valeur maximale de l'attribut *inter-octet-time-out* a été augmentée de 1 000 ms à 6 000 ms afin de permettre l'utilisation de supports de communication, lorsqu'il peut se produire de longs retards. La valeur par défaut a été changée à 25 ms pour l'aligner avec l'IEC 62056-46:2002/A1:2006, 6.4.4.3.4.

4.7.2.2 Description d'attribut

4.7.2.2.1 logical_name

Identifie l'instance de l'objet "IEC HDLC setup". Voir 6.2.20.

4.7.2.2.2 comm_speed

La vitesse de communication prise en charge par le port correspondant.

enum:

(0) 300 bauds;
(1) 600 bauds;
(2) 1 200 bauds;
(3) 2 400 bauds;
(4) 4 800 bauds;
(5) 9 600 bauds;
(6) 19 200 bauds;
(7) 38 400 bauds;
(8) 57 600 bauds;
(9) 115 200 bauds.

Cette vitesse de communication peut être neutralisée si le passage au mode HDLC d'un dispositif se fait par le biais d'un mode spécial d'un autre protocole.

4.7.2.2.3 window_size_transmit

Le nombre maximal de trames qu'un dispositif ou système peut émettre avant qu'il ait besoin de recevoir un acquittement provenant de la station correspondante. D'autres valeurs peuvent être négociées pendant l'ouverture de session.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.2.2.4 window_size_receive

Le nombre maximal de trames qu'un dispositif ou système peut recevoir avant qu'il ait besoin d'émettre un acquittement vers la station correspondante. D'autres valeurs peuvent être négociées pendant l'ouverture de session.

4.7.2.2.5 max_info_length_transmit

La longueur maximale du champ information qu'un dispositif peut émettre. Une valeur inférieure peut être négociée pendant l'ouverture de session.

4.7.2.2.6 max_info_length_receive

La longueur maximale du champ information qu'un dispositif peut recevoir. Une valeur inférieure peut être négociée pendant l'ouverture de session.

4.7.2.2.7 inter_octet_time_out

Définit le temps, en millisecondes, au-delà duquel, si aucun autre caractère n'est reçu en provenance de la station primaire, le dispositif traitera les données déjà reçues comme une trame complète.

4.7.2.2.8 inactivity_time_out

Définit le temps, en secondes, au-delà duquel, si aucune trame n'est reçue en provenance de la station primaire, le dispositif traitera une déconnexion.

Lorsque cette valeur est mise à 0, cela signifie que inactivity_time_out n'est pas opérationnel.

4.7.2.2.9 device_address

Contient l'adresse de dispositif physique d'un dispositif.

Dans le cas de l'adressage d'un octet:

0x00	NO_STATION Address;
0x01…0x0F	Réservé pour utilisation ultérieure;
0x10…0x7D	Espace d'adresses utilisable;
0x7E	Adresse de dispositif CALLING;
0x7F	Adresse de diffusion.

Dans le cas de l'adressage de deux octets:

0x0000	Adresse NO_STATION;
0x0001…0x000F	Réservé pour utilisation ultérieure;
0x0010…0x3FFD	Espace d'adresses utilisable;
0x3FFE	Adresse de dispositif physique CALLING;
0x3FFF	Adresse de diffusion.

4.7.3 IEC twisted pair (1) setup (class_id = 24, version = 1)

4.7.3.1 Généralités

Les instances de cette IC permettent de configurer l'échange de données sur le support à *paire torsadée avec signal de porteuse* comme indiqué dans l'IEC 62056-3-1:2013. Plusieurs voies de communication peuvent être configurées.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Le support de communication à *paire torsadée avec signal de porteuse* est largement utilisé dans le domaine des compteurs. L'utilisation de ce support présente l'avantage de faciliter l'installation et d'augmenter la fiabilité des communications grâce au signal de porteuse. Ce support peut être utilisé:

- entre les points d'accès au réseau local (LNAP) et les dispositifs finaux du compteur (interface M);
- entre les points d'accès au réseau local (LNAP) et les points d'accès aux réseaux de voisinage (NNAP);
- pour une connexion directe entre un HHU et le dispositif final du compteur.

L'IEC 62056-3-1:2013 spécifie trois profils de communication utilisant le support à paire torsadée avec signal de porteuse:

- sans DLMS®;
- avec DLMS®; et
- avec DLMS®/COSEM.

L'IEC 62056-31:1999 ne prend en charge que les deux premiers profils.

Le nouveau profil DLMS®/COSEM introduit une entité "Couche gestion du support" permettant l'initialisation du bus, la gestion des reconnaissances, la gestion des alarmes et la négociation de la vitesse de communication. Il permet également d'augmenter le débit jusqu'à 9 600 bauds. La couche transport prend en charge la segmentation et le réassemblage.

L'IC "IEC Twisted pair (1) set up" (class_id = 24, version = 0) prend en charge les deux premiers profils de communication spécifiés dans l'IEC 62056-31:1999.

La nouvelle version 1 prend en charge le profil DLMS®/COSEM. Avec cette introduction, l'utilisation de la version 0 est devenue obsolète.

L'utilisation des profils de communication spécifiés dans l'IEC 62056-3-1:2013 exige d'utiliser les services d'enregistrement fournis par l'Euridis Association: www.euridis.org.

Les objets d'interface COSEM suivants sont indispensables à l'établissement de l'échange de données sur le support à *paire torsadée avec signal de porteuse*:

- "IEC Twisted pair (1) setup": class_id = 24, version = 1;
- "MAC address": class_id = 43, version = 0;
- "Data": class_id = 1, version = 0.

Pour les codes OBIS, voir 6.2.21.

4.7.3.2 Vue d'ensemble

Les instances de cette IC permettent de configurer l'échange de données sur le support à *paire torsadée avec signal de porteuse* comme indiqué dans l'IEC 62056-3-1:2021. Plusieurs voies de communication peuvent être configurées.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IEC twisted pair (1) setup		0...n	class_id = 24, version = 1			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum	0		1	x + 0x08
3. comm_speed	(static)	enum	(2)	(7)	(2)	x + 0x10
4. primary_address_list	(static)	primary_address_list_type				x + 0x18
5. tabi_list	(static)	tabi_list_type				x + 0x20
Méthodes spécifiques		***o/f***				

4.7.3.3 Description d'attribut

4.7.3.3.1 logical_name

Identifie l'instance d'objet "IEC twisted pair setup". Voir 6.2.21.

4.7.3.3.2 mode

Cet attribut spécifie le mode de fonctionnement de cette interface.

enum:

(0) inactive. L'interface ignore toutes les trames reçues;
(1) toujours active;
(2) ... (127) réservé;
(128...250) spécifique au fabricant.

4.7.3.3.3 comm_speed

Contient la vitesse de communication prise en charge par le port.

enum:

(2) 1 200 bauds;
(3) 2 400 bauds;
(4) 4 800 bauds;
(5) 9 600 bauds;
(6) 19 200 bauds;
(7) 38 400 bauds.

NOTE L'IEC 62056-3-1:2013 prend en charge des débits compris entre 1 200 et 9 600 bauds.

4.7.3.3.4 primary_address_list

Contient la liste des adresses de stations primaires (ADP) pour lesquelles chaque dispositif logique de l'équipement réel (la station secondaire) a été programmé. Voir IEC 62056-3-1:2013, 5.2.4.

primary_address_list_type::= array unsigned

4.7.3.3.5 tabi_list

Représente la liste des TAB(i) pour lesquels l'équipement réel (la station secondaire) a été programmé dans le cas d'un appel de station oublié (voir IEC 62056-3-1:2013).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Lors de l'utilisation du profil de l'IEC 62056-3-1 avec DLMS®/COSEM, l'attribut tabi_list est composé d'un seul élément de valeur 0, qui est la valeur utilisée pour le processus de reconnaissance.

tabi_list_type::= array tabi_element

tabi_element: integer

4.7.3.4 MAC address

Les stations secondaires (en général des dispositifs finaux de compteur) contiennent une adresse de station secondaire (ADS). L'ADS doit être composée de 6 octets et des éléments présentés dans le Tableau 36. Cette adresse doit être unique dans le monde entier et doit être attribuée par le fabricant à la station secondaire. L'ADS attribuée est valide sur toute la durée de vie de la station secondaire.

Tableau 36 – Éléments de l'adresse ADS

Éléments de l'ADS	Description	Longueur (octet)	Type	Plage
manufacturer_id	Attribué au fabricant à la demande d'Euridis Association. Il doit être utilisé dans tous les dispositifs utilisant le support à *paire torsadée avec signal de porteuse* et fabriqué par ce fabricant.	1	BCD	0-99
year_of_manufacture	Contient l'année de fabrication du dispositif (les deux nombres inférieurs uniquement).	1		0-99
equipment_id	Se rapporte au type de matériel concerné et donne des informations relatives aux fonctions disponibles. Comme manufacturer_id, il est attribué par Euridis Association.	1		0-99
device_serial_number	Numéro de fabrication du dispositif attribué par le fabricant. Il convient que sa valeur soit identique à celle contenue par l'objet Device ID 1 (voir IEC 62056-6-1:2023, Tableau 8).	3		000001-999999 000000 est réservé
EXEMPLE 031267123456: 03: ITRON International; 12: Année 2012; 67: Compteur intelligent LINKY monophasé; 123456: Numéro de série commençant chaque année à partir de 000001.				

4.7.3.5 Registre des erreurs fatales

Chaque dispositif mettant en œuvre le profil de communication DLMS®/COSEM spécifié dans l'IEC 62056-3-1:2013 doit fournir un registre des erreurs contenant le résultat de la dernière communication avec la station primaire. La structure du registre des erreurs fatales doit être conforme au Tableau 37.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 37 – Registre des erreurs fatales

Réf	Nom	Description
Bit 0	EP-3F	Erreur de transmission. Le TOE d'expiration du délai s'est écoulé sans envoi d'octet, donnant lieu à une impossibilité d'envoyer le reste de la trame.
Bit 1	EP-4F	Erreur de réception. Le nombre d'octets reçus est supérieur à la valeur maximale prévue.
Bit 2	EP-5F	Délai TARSO écoulé en recevant une trame RSO. Ne concerne pas les stations secondaires (serveur). Concerne la station primaire seulement (client).
Bit 3	EL-1F	Indication d'alarme reçue pendant une association. Non pertinent. Concerne la station primaire seulement (client).
Bit 4	EL-2F	Réponse incorrecte de la station secondaire après transmissions répétées MaxRetry d'une demande.
Bit 5	EA-1F	TAB incorrect depuis le serveur. Ne concerne pas les stations secondaires (serveur). Concerne la station primaire seulement (client).
Bit 6	EA-2F	Erreur d'authentification sur les données reçues du serveur. Ne concerne pas les stations secondaires (serveur). Concerne la station primaire seulement (client).
Bit 7	EA-3F	Erreur d'authentification détectée par la station secondaire.

4.7.4 Modem configuration (class_id = 27, version = 1)

4.7.4.1 Vue d'ensemble

Cette IC permet de modéliser la configuration et l'initialisation des modems utilisés pour le transfert de données en provenance/à destination d'un dispositif. Plusieurs modems peuvent être configurés.

Modem configuration		**0...n**	**class_id = 27, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	9	5	x + 0x08
3. initialization_string	(static)	array				x + 0x10
4. modem_profile	(static)	array				x + 0x18
Méthodes spécifiques		***o/f***				

4.7.4.2 Description d'attribut

4.7.4.2.1 logical_name

Identifie l'instance de l'objet "Modem configuration". Voir 6.2.6.

4.7.4.2.2 comm_speed

La vitesse de communication entre le dispositif et le modem, pas nécessairement la vitesse de communication sur le réseau étendu.

enum:

300 bauds;
600 bauds;
1 200 bauds;
2 400 bauds;
4 800 bauds;
9 600 bauds;
19 200 bauds;
38 400 bauds;
57 600 bauds;
115 200 bauds.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.4.2.3 initialization_string

Contient toutes les commandes d'initialisation nécessaires devant être envoyées au modem afin de le configurer correctement. Cela peut inclure la configuration de caractéristiques spéciales du modem.

```
array        initialization_string_element

initialization_string_element::= structure
{
request:                    octet-string,
response:                   octet-string,
delay_after_response:       long-unsigned
}
```

Si le tableau contient plus d'un initialization_string_element, les demandes peuvent être envoyées en séquence. La demande suivante est envoyée après la réponse attendue correspondant à la demande précédente et après une attente d'un temps delay_after_response [ms], pour permettre au modem d'exécuter la demande.

NOTE Il est pris pour hypothèse que le modem est préconfiguré et accepte, de ce fait, la chaîne d'initialisation (initialization_string). Si l'initialisation n'est pas nécessaire, la chaîne d'initialisation est vide.

4.7.4.2.4 modem_profile

Définit la mise en correspondance des commandes/réponses de la norme Hayes avec les chaînes spécifiques au modem.

```
array        modem_profile_element

modem_profile_element: octet-string
```

Le tableau modem_profile doit contenir les chaînes correspondantes pour le modem utilisé dans l'ordre suivant:

```
élément 0:      OK;
élément 1:      CONNECT;
élément 2:      RING;
élément 3:      NO CARRIER;
élément 4:      ERROR;
élément 5:      CONNECT 1 200;
élément 6       NO DIAL TONE;
élément 7:      BUSY;
élément 8:      NO ANSWER;
élément 9:      CONNECT 600;
élément 10:     CONNECT 2 400;
élément 11:     CONNECT 4 800;
élément 12      CONNECT 9 600;
élément 13:     CONNECT 14 400;
élément 14:     CONNECT 28 800;
élément 15:     CONNECT 33 600;
élément 16:     CONNECT 56 000.
```

4.7.5 Auto answer (class_id = 28, version = 2)

4.7.5.1 Généralités

NOTE 1 La version 1 de la classe "Auto answer" était une version provisoire.

La version 0 de la classe "Auto answer" modélise la manière dont le dispositif gère les appels entrants pour demander la connexion du modem.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Dans la version 2, de nouvelles capacités sont ajoutées pour gérer les demandes de réactivation qui peuvent se présenter sous la forme d'un appel ou d'un message de réactivation (un message SMS [vide], par exemple). A la suite d'une demande de réactivation réussie, le dispositif se connecte au réseau. Voir aussi Annexe A.

Pour les deux fonctions, une sécurité supplémentaire est assurée en ajoutant la possibilité de vérification du numéro appelant: les appels ou les messages sont acceptés uniquement en fonction d'une liste préalablement définie d'appelants. Cette fonction exige la présence d'un service d'identification de la ligne appelante (CLI) dans le réseau de communication utilisé.

NOTE 2 Le processus de réactivation est totalement découplé des services AL, c'est-à-dire qu'un message de réactivation ne peut pas contenir de demandes de service xDLMS. Il s'agit d'éviter la création d'une porte dérobée. Les messages xDLMS peuvent être échangés dans les messages SMS à l'issue du processus de réactivation.

Auto answer		**0...n**	**class_id = 28, version = 2**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. listening_window	(static)	array				x + 0x10
4. status	(dyn.)	enum				x + 0x18
5. number_of_calls	(static)	unsigned				x + 0x20
6. number_of_rings	(static)	nr_rings_type				x + 0x28
7. list_of_allowed_callers	(static)	array				x + 0x30
Méthodes spécifiques		***o/f***				

4.7.5.2 Description d'attribut

4.7.5.2.1 logical_name

Identifie l'instance de l'objet "Auto answer". Voir 6.2.6.

4.7.5.2.2 mode

Définit le mode de fonctionnement de la ligne lorsque le dispositif est en mode de réponse automatique.

enum:

(0) ligne dédiée au dispositif;
(1) gestion de ligne partagée avec un nombre limité d'appels autorisés. Une fois que le nombre d'appels est atteint, l'état de la fenêtre devient "inactif" jusqu'à la prochaine date de début, quel que soit le résultat de l'appel;
(2) gestion de ligne partagée avec un nombre limité d'appels réussis autorisés. Une fois que le nombre de communications réussies est atteint, l'état de la fenêtre devient "inactif" jusqu'à la prochaine date de début;
(3) actuellement aucun modem connecté;
(200…255) modes spécifiques au fabricant.

4.7.5.2.3 listening_window

Définit les moments où la fenêtre de communication devient active (start_time) et inactive (end_time). start_time définit implicitement la période.

EXEMPLE Lorsque le jour du mois n'est pas spécifié (égal à 0xFF), cela signifie qu'il s'agit de la gestion d'une fenêtre d'écoute quotidienne. Une gestion de fenêtre quotidienne, mensuelle, etc. peut être définie.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
array        window_element

window_element::= structure
{
start_time:  octet-string,
end_time:    octet-string
}
```

start_time et end_time sont formatés comme spécifié en 4.1.6.1 pour *date-time*.

4.7.5.2.4 status

Définit l'état de la fenêtre.

enum:

(0) Inactive: le dispositif ne gérera aucun nouvel appel entrant. Cet état est automatiquement réinitialisé à "Active" lorsque la fenêtre d'écoute suivante commence;
(1) Active: le dispositif peut répondre au nouvel appel entrant;
(2) Locked: cette valeur peut être fixée automatiquement par le dispositif ou par un client spécifique lorsque ce client a terminé sa session de lecture et souhaite rendre la ligne au client avant la fin de la durée de la fenêtre. Cet état est automatiquement réinitialisé à "Active" lorsque la fenêtre d'écoute suivante commence.

4.7.5.2.5 number_of_calls

Ce nombre est la référence utilisée dans les modes 1 et 2.

Lorsqu'il est mis à 0, cela signifie qu'il n'y a pas de limite.

4.7.5.2.6 number_of_rings

Définit le nombre de sonneries avant que le compteur ne connecte le modem. Deux cas sont distingués: le nombre de sonneries dans la fenêtre définie par l'attribut *listening_window* et le nombre de sonneries à l'extérieur de la *listening_window*.

```
nr_rings_type::= structure
{
    nr_rings_in_window:      unsigned (0 = pas de connexion dans la fenêtre),
    nr_rings_out_of_window: unsigned (0 = pas de connexion hors de la fenêtre)
}
```

Si le nombre de sonneries à l'intérieur et à l'extérieur de la fenêtre est le même, le modem se connecte toujours, quels que soient les paramètres de *listening_window*.

4.7.5.2.7 list_of_allowed_callers

Contient une liste (facultative) de numéros appelants qui limite encore plus la connectivité du modem selon le numéro appelant. Il contrôle également l'acceptation des appels ou messages de réactivation (SMS, par exemple) provenant d'un numéro appelant.

Cela exige la présence d'un service d'identification de la ligne appelante (CLI) dans le réseau de communication utilisé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
list_of_allowed_callers::= array list_of_allowed_callers_element

list_of_allowed_callers_element::= structure
{
      caller_id:   octet-string,
      call_type:   enum
}
```

– l'élément caller_id contient un numéro appelant à partir duquel les appels ou messages (SMS, par exemple) sont acceptés. Les caractères génériques "?" et "*" sont pris en charge. Avec "?", un seul caractère correspond. Avec "*", toutes les chaînes de caractères correspondent. "*" peut uniquement être utilisé au début ou à la fin d'un numéro, mais jamais au milieu, ni seul.

EXEMPLE 1: "+994193500" = seuls les appels provenant du numéro "+994193500" sont acceptés.

EXEMPLE 2: "+9941935????" = les appels provenant de tous les numéros compris entre "+99419350000" et "+99419359999" sont acceptés.

EXEMPLE 3: "7777*" = les appels provenant de tous les numéros commençant par "7777" sont acceptés.

EXEMPLE 4: "*9000" = les appels provenant de tous les numéros se terminant par "9000" sont acceptés.

– l'élément call_type définit l'objet de l'appel, c'est-à-dire s'il s'agit d'un appel CSD normalisé ou d'un appel/message de réactivation.

enum:

(0) = appel CSD normal; le modem se connecte uniquement si le numéro appelant correspond à une entrée de la liste. Il est soumis à essai en plus de tous les autres attributs (*number_of_rings, listening_windows*, par exemple).

(1) = requête de réactivation; les appels ou messages provenant de ce numéro appelant sont gérés comme des requêtes de réactivation. La requête de réactivation est immédiatement traitée, quels que soient les autres attributs tels que *number_of_rings* et *listening_window* (sauf si le numéro appelant est également présent dans la liste des appels CSD normaux; voir ci-dessous).

Le message reçu doit être complètement vide. Sinon, il n'est pas traité comme un message de réactivation.

Si le message contient une APDU xDLMS valide provenant d'un client d'une AA préalablement établie, le service xDLMS correspondant est exécuté à la place du traitement de la requête de réactivation.

Si le message n'est pas vide, mais qu'il ne contient pas d'APDU xDLMS valide provenant d'un client d'une AA préalablement établie, le serveur ne réagit pas.

Pour un appel de type (1), le modem *ne se connecte pas*, quel que soit le résultat du processus de réactivation.

Pour un appel de type (1) et de type (0): voir ci-dessous.

Si le même numéro appelant est défini comme étant à l'origine d'un appel CSD normal [type (0)] et d'une requête de réactivation [type (1)], la règle suivante s'applique pour les appels entrants: la requête de réactivation est uniquement traitée si l'appelant déconnecte la ligne avant d'atteindre le critère *number_of_rings* (en fonction de *listening_window*). Si le critère *number_of_rings* est satisfait, une connexion modem est établie.

La valeur du paramètre *number_of_rings* doit être suffisamment élevée pour permettre à l'initiateur de l'appel de contrôler le comportement du récepteur de l'appel sans connaître les instances temporelles des sonneries côté récepteur.

NOTE Si *list_of_allowed_callers* est vide (= tableau [0]), la fonction de réponse automatique fonctionne dans le type (0) = appel CSD normal, et le modem se connecte quel que soit le numéro appelant.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.6 Auto connect (class_id = 29, version = 2)

4.7.6.1 Vue d'ensemble

La version 1 de la classe "Auto connect" modélise la manière dont le dispositif procède à la numérotation automatique ou à l'envoi des messages à l'aide de différents services.

Dans la version 2, de nouvelles capacités ont été ajoutées afin de modéliser la connexion du dispositif à un réseau de communication. La connexion au réseau peut être permanente, à l'intérieur d'une fenêtre de temps ou à l'appel de la méthode de connexion.

Auto connect		**0...n**	**class_id = 29, version = 2**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mode	(static)	enum				x + 0x08
3. repetitions	(static)	unsigned				x + 0x10
4. repetition_delay	(static)	long-unsigned				x + 0x18
5. calling_window	(static)	array				x + 0x20
6. destination_list	(static)	array				x + 0x28
Méthodes spécifiques		***o/f***				
1. connect (data)		f				x + 0x30

4.7.6.2 Description d'attribut

4.7.6.2.1 logical_name

Identifie l'instance de l'objet "Auto connect". Voir 6.2.6.

4.7.6.2.2 mode

Commande la fonction de connexion automatique en termes de synchronisation, de type de message et d'infrastructure à utiliser.

Les modes (1) à (3) sont dédiés aux services CSD.

Les modes (4) à (6) sont dédiés à l'envoi de messages particuliers à l'aide d'une infrastructure particulière.

Les modes (101) à (104) s'appliquent aux connexions de réseau à commutation de paquets uniquement (GPRS, par exemple).

enum:

(0) pas de connexion automatique. Le dispositif ne se connecte jamais,
(1) numérotation automatique autorisée à tout moment, les valeurs définies dans *calling_window* sont ignorées;
(2) numérotation automatique autorisée dans la limite du temps de validité de *calling_window*;
(3) numérotation automatique "régulière" autorisée dans la limite du temps de validité de *calling_window*; numérotation automatique déclenchée par une "alarme" autorisée à tout moment;
(4) envoi de SMS via les réseaux mobiles terrestres publics (PLMN);
(5) envoi de SMS via le PSTN;
(6) envoi de courriel (email);
(7...99) réservé;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(101) le dispositif est connecté en permanence au réseau de communication;
(102) le dispositif est connecté en permanence au réseau de communication pendant la durée de validité de la fenêtre d'appel. Le dispositif est déconnecté hors de la fenêtre d'appel. Aucune connexion possible hors de la fenêtre d'appel;
(103) le dispositif est connecté en permanence au réseau de communication pendant la durée de validité de la fenêtre d'appel. Le dispositif est déconnecté hors de la fenêtre d'appel, mais il se connecte au réseau de communication dès que la méthode de connexion est appelée;
(104) le dispositif est en général déconnecté. Il se connecte au réseau de communication dès que la méthode de connexion est appelée;
(105…199) réservé;
(200…255) modes spécifiques au fabricant.

4.7.6.2.3 repetitions

Nombre maximal de nouvelles tentatives, en cas de tentatives de connexion non réussies.

4.7.6.2.4 repetition_delay

Délai, en secondes, à observer avant qu'une tentative de connexion infructueuse puisse être répétée.

repetition_delay = 0 signifie que le retard n'est pas spécifié.

4.7.6.2.5 calling_window

Contient les points dans le temps auxquels la fenêtre devient active (start_time) et inactive (end_time). start_time définit implicitement la période.

EXEMPLE Lorsque le jour du mois n'est pas spécifié (égal à 0xFF), cela signifie que la fenêtre d'appel est gérée quotidiennement. Une gestion de fenêtre quotidienne, mensuelle, etc. peut être définie.

```
array        window_element

window_element::= structure
{
start_time:  octet-string,
end_time:    octet-string
}
```

start_time et end_time sont formatés comme spécifié en 4.1.6.1 pour *date-time*.

4.7.6.2.6 destination_list

Contient la liste de destinations (par exemple numéros de téléphone, adresses email ou leurs combinaisons) vers lesquelles le ou les messages doivent être envoyés dans certaines conditions. Les conditions et leur lien aux éléments du tableau ne sont pas définis ici.

```
array        destination

destination::= octet-string
```

4.7.6.3 Description de la méthode

4.7.6.4 connect (data)

Lance le processus de connexion au réseau de communication conformément aux règles définies par l'intermédiaire de l'attribut de mode.

```
data::= integer (0)
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.7 GPRS modem setup (class_id = 45, version = 0)

4.7.7.1 Vue d'ensemble

Cette IC permet d'installer des modems GPRS en gérant toutes les données nécessaires à la gestion du modem.

GPRS modem setup		0...n	class_id = 45, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. APN	(static)	octet-string				x + 0x08
3. PIN_code	(static)	long-unsigned				x + 0x10
4. quality_of_service	(static)	structure				x + 0x18
Méthodes spécifiques		***o/f***				

4.7.7.2 Description d'attribut

4.7.7.2.1 logical_name

Identifie l'instance de l'objet "GPRS modem setup". Voir 6.2.23.

4.7.7.2.2 APN

Définit le nom de point d'accès du réseau.

4.7.7.2.3 PIN_code

Contient le numéro d'identification personnel.

4.7.7.2.4 quality_of_service

Spécifie les paramètres de qualité de service. Il s'agit d'une structure à 2 éléments:

– le premier élément définit les caractéristiques par défaut ou minimales du réseau concerné. Ces paramètres doivent être réglés à la valeur de meilleur effort;

– le second élément définit les paramètres demandés.

```
quality_of_service::= structure

{
default:            qos_element,
requested:          qos_element
}
qos_element::= structure
{
precedence:             unsigned,
delay:                  unsigned,
reliability:            unsigned,
peak throughput:        unsigned,
mean throughput:        unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.8 GSM diagnostic (class_id: 47, version: 2)

4.7.8.1 Vue d'ensemble

Le réseau cellulaire fait l'objet de modifications constantes en termes d'état d'enregistrement, de qualité du signal, etc. Il est nécessaire de surveiller et de journaliser les paramètres pertinents afin d'obtenir des informations de diagnostic permettant d'identifier des problèmes de communication dans le réseau.

Une instance de la classe "GSM diagnostic" enregistre les paramètres du réseau GSM/GPRS, UMTS, CDMA ou LTE nécessaires à l'analyse du fonctionnement du réseau.

Un objet "Profile generic" de diagnostic GSM est également disponible pour capturer les attributs de l'objet "GSM diagnostic" (voir IEC 62056-6-1:2023, 6.5).

GSM diagnostic		**0...n**	**class_id = 47, version = 2**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. operator	(dyn.)	visible-string				x + 0x08
3. status	(dyn.)	enum	0	255	0	x + 0x10
4. cs_attachment	(dyn.)	enum	0	255	0	x + 0x18
5. ps_status	(dyn)	enum	0	255	0	x + 0x20
6. cell_info	(dyn.)	cell_info_type				x + 0x30
7. adjacent_cells	(dyn.)	array				x + 0x38
8. capture_time	(dyn.)	date-time				x + 0x40
Méthodes spécifiques		***o/f***				

4.7.8.2 Description d'attribut

4.7.8.2.1 logical_name

Identifie l'instance de l'objet "GSM Diagnostic". Pour les noms logiques, voir 6.2.23.

4.7.8.2.2 operator

Contient le nom de l'opérateur du réseau ("YourNetOp", par exemple).

4.7.8.2.3 status

Indique l'état d'enregistrement du modem.

enum:

(0) non enregistré;
(1) enregistré, réseau domestique;
(2) non enregistré, mais la terminaison mobile (MT) recherche actuellement un nouvel opérateur auprès duquel s'enregistrer;
(3) enregistrement refusé;
(4) inconnu;
(5) enregistré, itinérance;
(6)... (255) réservé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.8.2.4 cs_attachment

Indique l'état commuté du circuit en cours.

enum:

(0) inactif;
(1) appel entrant;
(2) actif;
(3)…(255) réservé.

4.7.8.2.5 ps_status

Le champ de valeur *ps_status* indique l'état commuté par paquets du modem.

enum:

(0) inactif;
(1) GPRS;
(2) EDGE;
(3) UMTS,
(4) HSDPA;
(5) LTE;
(6) CDMA;
(7) LTE Cat M1;
(8) LTE Cat NB1;
(9) LTE Cat NB2;
(10) …(255) réservé.

4.7.8.2.6 cell_info

Représente les informations de cellule:

```
cell_info_type::= structure
{
cell_ID:              double-long-unsigned,
location_ID:          long-unsigned,
signal_strength:      unsigned,
ber:                  unsigned,
mcc:                  long-unsigned,
mnc:                  long-unsigned,
channel_number:       double-long-unsigned
}
```

Où:

– cell_ID: ID de cellule à quatre octets au format hexadécimal;

– location_ID: Code de zone locale (LAC) à deux octets dans le cas des réseaux GSM ou code de zone de suivi (TAC) dans le cas des réseaux UMTS, CDMA ou LTE au format hexadécimal ("00C3" équivaut à 195 en notation décimale, par exemple);

– signal_strength: Représente la qualité du signal:

(0) –113 dBm au maximum;
(1) –111 dBm;
(2…30) -109…-53 dBm;
(31) –51 au moins;
(99) inconnue ou indétectable;

– ber: Taux d'erreur sur les bits (BER) en pourcentage:

(0…7) comme les valeurs RXQUAL_n spécifiées dans l'ETSI GSM 05.08:1996, 8.2.4
(99) inconnue ou indétectable;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– mcc: indicatif de pays du mobile (MCC) du réseau de service, tel que défini dans l'UIT-T E.212 (05.2008);
– mnc: code de réseau du mobile (MNC) du réseau de desserte, tel que défini dans l'UIT-T E.212 (05.2008);
– channel_number: représente le numéro de canal de fréquence radio absolu (ARFCN ou EARFCN pour le réseau LTE).

4.7.8.2.7 adjacent_cells

```
array            adjacent_cell_info

adjacent_cell_info::= structure
{
    cell_ID:          double-long-unsigned,
    signal_strength:  unsigned
}
```

Où:

– cell_ID: ID de cellule à quatre octets au format hexadécimal;
– signal_strength: Représente la qualité du signal:

(0) –113 dBm au maximum;
(1) –111 dBm;
(2...30) -109...-53 dBm;
(31) –51 au moins;
(99) inconnue ou indétectable.

4.7.8.2.8 capture_time

Contient la date et l'heure de capture des données.

date-time est mis en forme comme spécifié en 4.1.6.1.

4.7.9 LTE monitoring (class_id: 151, version: 1)

4.7.9.1 Vue d'ensemble

Les instances de l'IC "LTE monitoring" permettent de surveiller les modems LTE en gérant toutes les données nécessaires à cet effet.

La version 1 de l'IC "LTE monitoring" couvre également les réseaux LTE Cat M1 et NB-IoT.

LTE monitoring		**0...n**	**class_id = 151, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. LTE_network_parameters	(dyn.)	LTE_network_ parameters_type				x + 0x08
3. LTE_quality_of_service	(dyn)	LTE_qos_type				x + 0x10
Méthodes spécifiques		***o/f***				

4.7.9.2 Description d'attribut

4.7.9.2.1 logical_name

Identifie l'instance de l'objet "LTE monitoring". Voir 6.2.23.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.9.2.2 LTE_network_parameters

Représente les paramètres du réseau LTE.

```
LTE_network_parameters_type::= structure

{
T3402:              long-unsigned,
T3412:              long-unsigned,
T3412ext2:          double-long-unsigned,
T3324               long-unsigned,
TeDRX               double-long unsigned,
TPTW                long-unsigned,
qRxlevMin:          integer,
qRxlevMinCE-r13:    integer,
qRxlevMinCE1-r13    integer
}
```

Où:

– T3402: temporisateur en secondes, utilisé dans la procédure de sélection PLMN, et envoyé au modem par le réseau. Voir 3GPP TS 24.301 V13.11.0 (2018-01) pour plus de détails;

– T3412: temporisateur en secondes utilisé pour gérer la procédure de mise à jour de la zone de suivi périodique, et envoyé au modem par le réseau. Voir 3GPP TS 24.301 V13.11.0 (2018-01) pour plus de détails;

– T3412ext2: temporisateur en secondes (temporisateur de mise à jour de la zone de suivi périodique étendue). Voir 3GPP TS 24.301 V13.11.0 (2018-01) et 3GPP TS 24.008 V13.7.0 (2016-10) pour plus de détails;

– T3324: temporisateur en secondes (temporisateur d'activation du mode d'économie d'énergie). Voir 3GPP TS 24.301 V13.11.0 (2018-01) et 3GPP TS 24.008 V13.7.0 (2016-10) pour plus de détails;

– TeDRX: temporisateur (temporisateur de cycle DRX en mode inactif étendu). Voir 3GPP TS 24.301 V13.11.0 (2018-01) et 3GPP TS 24.008 V13.7.0 (2016-10) pour plus de détails;

– La valeur double-long unsigned value doit être multipliée par 0,01 pour obtenir la valeur réelle en secondes. Par exemple, 512 représente 5,12 secondes;

– TPTW: temporisateur (fenêtre de temps de radiomessagerie DRX en mode inactif étendu). Voir 3GPP TS 24.301 V13.11.0 (2018-01) et 3GPP TS 24.008 V13.7.0 (2016-10) pour plus de détails;

– La valeur double-long unsigned doit être multipliée par 0,01 pour obtenir la valeur réelle en secondes. Par exemple, 512 représente 5,12 secondes;

– qRxlevMin: niveau Rx minimal exigé dans la cellule en dBm tel que défini dans 3GPP TS 36.304 V13.8.0 (2018-01);

– qRxlevMinCE-r13: niveau Rx minimal exigé en mode A d'amélioration de la couverture (CE), LTE Cat M1. Pour ce mode, une valeur de -70 à -22 par échelon de 1 est exigée;

NOTE 1 Ce champ n'est pas utilisé dans le cas d'un réseau LTE Cat NB1 ou NB2.

– qRxlevMinCE1-r13: niveau Rx minimal exigé en mode B d'amélioration de la couverture (CE), LTE Cat M1. Pour ce mode, une valeur de -78 à -22 par échelon de 1 est exigée.

NOTE 2 Ce champ n'est pas utilisé dans le cas d'un réseau LTE Cat NB1 ou NB2.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.7.9.2.3 LTE_quality_of_service

Représente la qualité de service du réseau LTE

```
LTE_QoS_type::= structure
{
(N)RSRQ:                integer,
(N)RSRP:                integer,
SNR:                    integer,
Coverage Enhancement:   enum
}
```

Où:

(N)RSRQ représente la qualité du signal telle que définie dans 3GPP TS 36.133. L'utilisation de ce paramètre peut être déterminée à partir de l'attribut ps_status de l'IC "GSM diagnostic":

- pour LTE Cat M1, une plage de valeurs comprise entre -30 et 46 est nécessaire pour représenter RSRQ. Voir 3GPP TS 36.133 V13.11.0 (2018-04) pour plus de détails;
- pour LTE Cat NB1 et LTE Cat NB2, une plage de valeurs comprise entre -30 et 46 est nécessaire pour représenter NRSRQ. Voir 3GPP TS 36.133 V14.4.0 (2017-07) pour plus de détails.

(-30)	-34 dB,
(-29)	-33,5 dB,
(-28)..(-1)	-33 dB, -19,5 dB,
(0)	-19,5 dB ou moins,
(1)	-19 dB,
(2)..(31)	-18,5 dB .. -4 dB,
(32)	-3,5 dB,
(33)	-3 dB,
(34)	-3 dB au maximum,
(35)	-2,5 dB,
(36)	-2 dB,
(37)..(45)	-1,5 dB .+2,5 dB,
(46)	2,5 dB au minimum,
(99)	inconnnu ou indétectable,
autres valeurs	réservé

(N)RSRP représente le niveau de signal tel que défini dans 3GPP TS 36.133. L'utilisation de ce paramètre est déterminée à partir de l'attribut ps_status de l'IC "GSM diagnostic":

- pour ps_status = 7 (LTE Cat M1), une plage de valeurs comprise entre -17 et 97 est nécessaire pour représenter RSRP. Voir 3GPP TS 36.133 V13.11.0 (2018-04) pour plus de détails;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

(-17)	-156 dBm,
(-16)	-155 dBm,
(-15)..(-1)	-154 dBm .. -140 dBm,
(0)	-140 dBm au maximum,
(1)	-139 dB,
(2)	-138 dBm,
(3)..(96)	-137 dBm .. -44 dBm,
(97)	-44 dBm au minimum,
(127)	inconnnu ou indétectable,
autres valeurs	réservé

NOTE 1 Selon l'utilisation des modes d'amélioration de la couverture (CE), une valeur déclarée de (0) est utilisée ou non (pour les dispositifs qui n'utilisent pas de modes CE, les valeurs [-17..-1] ne sont pas utilisées). Une valeur déclarée de (0) signifie dans ce cas une RSRP < -140 dBm).

- pour ps_status = 8 ou 9 (LTE Cat NB1 ou LTE Cat NB2), une plage de valeurs comprise entre 0 et 113 est nécessaire pour représenter la valeur NRSRP. Voir 3GPP TS 36.133 V14.4.0 (2017-07) pour plus de détails.

(0)	-156 dBm,
(1)	-155 dBm,
(2)	-154 dBm
(3)..(112)	-153 dBm .. -44 dBm,
(113)	-44 dBm au minimum,
(127)	inconnnu ou indétectable,
autres valeurs	réservé

NOTE 2 Pour la valeur NRSRP (NB-IoT), la mise en correspondance est différente de celle de la valeur RSRP (LTE Cat M1).

- SNR représente le rapport signal/bruit dans une plage comprise entre -20 dB et 50 dB. Voir 3GPP TS 36.101 V15.4.0 (2019-01) pour plus de détails.
- Coverage Enhancement représente le mode d'amélioration de la couverture A ou B dans le cas de LTE Cat M1, Niveau de CE 0, 1 ou 2 dans le cas de LTE Cat NB1/NB2. Voir 3GPP TS 36.331 V15.5.1 (2019-05), 3GPP TS 36.321 V15.5.0 (2019-05) et 3GPP TS 36.213 V15.5.0 (2019-05) pour plus de détails.

Enum	LTE Cat M1	NB-IoT
(0)	Mode CE A	Niveau CE 0
(1)	Mode CE B	Niveau CE 1
(2)	(non utilisé)	Niveau CE 2
autres valeurs	réservé	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8 Classes d'interfaces pour l'établissement d'échange de données via M-Bus

4.8.1 Vue d'ensemble

Les classes d'interfaces M-Bus spécifiées dans le présent paragraphe sont utilisées dans le cadre de deux scénarios distincts:

a) un serveur DLMS®/COSEM hébergé par un M-Bus maître et échangeant des APDU M-Bus dédiées avec les M-Bus esclaves;

b) un client DLMS®/COSEM hébergé par un M-Bus maître et échangeant des APDU DLMS®/COSEM avec les serveurs DLMS®/COSEM hébergés par les M-Bus esclaves.

Dans le cas a), les instances des classes d'interface M-Bus suivantes sont utilisées pour établir et gérer le support M-Bus dans le serveur DLMS®/COSEM:

- M-Bus client (class_id = 72) (voir 4.8.3);
- M-Bus master port setup (class_id = 74) (voir 4.8.5);
- M-Bus diagnostic (class_id = 77, version = 0) (voir 4.8.7).

Dans le cas b), les instances des classes d'interfaces M-Bus suivantes sont utilisées dans le serveur DLMS®/COSEM:

- DLMS®/COSEM server M-Bus port setup (class_id = 76) (voir 4.8.6);
- M-Bus slave port setup (class_id = 25) (voir 4.8.2);
- Wireless Mode Q channel (class_id = 73) (voir 4.8.4);
- M-Bus diagnostic (class_id = 77, version = 0) (voir 4.8.7).

4.8.2 M-Bus slave port setup (class_id = 25, version = 0)

4.8.2.1 Vue d'ensemble

NOTE Le nom de cette IC a été changé de "M-BUS port setup" en "M-Bus slave port setup" afin d'indiquer qu'elle sert à établir l'échange de données lorsqu'un serveur COSEM communique avec un client COSEM en utilisant le M-Bus câblé.

Cette IC permet de modéliser et de configurer des voies de communication selon l'EN 13757-2:2004. Plusieurs voies de communication peuvent être configurées.

M-Bus slave port setup	**0...n**	**class_id = 25, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. default_baud (static)	enum	0	5	0	x + 0x08
3. avail_baud (static)	enum	0	7		x + 0x10
4. addr_state (static)	enum				x + 0x18
5. bus_address (static)	unsigned				x + 0x20
Méthodes spécifiques	***o/f***				

4.8.2.2 Description d'attribut

4.8.2.2.1 logical_name

Identifie l'instance de l'objet "M-Bus slave port setup". Voir 6.2.22.

4.8.2.2.2 default_baud

Définit le débit en bauds pour la séquence d'ouverture.

enum:

(0) 300 bauds;
(3) 2 400 bauds;
(5) 9 600 bauds.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.2.2.3 avail_baud

Définit les débits en bauds qui peuvent être négociés après démarrage.

enum:

(0) 300 bauds;
(1) 600 bauds;
(2) 1 200 bauds;
(3) 2 400 bauds;
(4) 4 800 bauds;
(5) 9 600 bauds;
(6) 19 200 bauds;
(7) 38 400 bauds.

4.8.2.2.4 addr_state

Définit si une adresse a été ou non attribuée au dispositif depuis la dernière mise sous tension du dispositif.

enum:

(0) n'a pas encore eu d'adresse attribuée;
(1) a une adresse attribuée, soit par réglage manuel, soit par une méthode automatisée.

4.8.2.2.5 bus_address

L'adresse actuellement attribuée sur le bus pour le dispositif.

NOTE Si aucune adresse de bus n'est attribuée, la valeur est 0.

4.8.3 M-Bus client (class_id = 72, version = 1)

4.8.3.1 Vue d'ensemble

Les instances de "M-Bus client" permettent d'installer des dispositifs M-Bus esclaves en utilisant le M-Bus câblé et d'échanger des données avec eux. Chaque objet "M-Bus client" commande un dispositif M-Bus esclave. Pour obtenir des détails sur la couche application M-Bus dédiée, voir EN 13757-3:2013.

NOTE La version 1 de l'IC "M-Bus client" est conforme à l'EN 13757-3:2013.

Le dispositif M-Bus client peut avoir une ou plusieurs interfaces physiques M-Bus, qui peuvent être configurées à l'aide d'instances de l'IC "M-Bus master port setup" (voir 4.8.5).

Un dispositif M-Bus esclave est identifié avec son adresse primaire, son numéro d'identification, son identifiant de fabricant, etc. comme défini dans l'EN 13757-3:2013, Article 5 (Envoi et réponse de données variables). Ces paramètres sont contenus dans les attributs respectifs de l'IC "M-Bus client".

Les valeurs à capturer à partir d'un dispositif M-Bus esclave sont identifiées par l'attribut *capture_definition*, contenant une liste d'identifiants de données (DIB, VIB) pour le dispositif M-Bus esclave. Les données sont capturées périodiquement ou sur un déclencheur approprié. Chaque élément de donnée est stocké dans un objet "M-Bus value" de l'IC "Extended register". Les objets "M-Bus value" peuvent être capturés dans les objets M-Bus "Profile generic", éventuellement avec d'autres objets, non spécifiques à M-Bus.

En utilisant les méthodes des objets "M-Bus client", les dispositifs M-Bus esclaves peuvent être installés et désinstallés.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Il est également possible d'envoyer des données à des dispositifs M-Bus esclaves et d'exécuter des opérations telles que la réinitialisation d'alarmes, la synchronisation d'horloge, le transfert d'une clé de cryptage, etc.

Le champ de configuration défini dans l'EN 13757-3:2013, 5.12 donne des informations sur le mode de cryptage et le nombre d'octets cryptés.

L'état de la clé de cryptage indique si la clé a été définie, transférée vers le dispositif M-Bus esclave et utilisée avec ce dernier.

M-Bus client		**0...n**	**class_id = 72, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	***Nom court***
1. logical_name	(static)	octet-string				x
2. mbus_port_reference	(static)	octet-string				x + 0x08
3. capture_definition	(static)	array			vide	x + 0x10
4. capture_period	(static)	double-long-unsigned			0	x + 0x18
5. primary_address		unsigned			0	x + 0x20
6. identification_number	(dyn.)	double-long-unsigned			0	x + 0x28
7. manufacturer_id	(dyn.)	long-unsigned			0	x + 0x30
8. version	(dyn.)	unsigned			0	x + 0x38
9. device_type	(dyn.)	unsigned			0	x + 0x40
10. access_number	(dyn.)	unsigned			0	x + 0x48
11. status	(dyn.)	unsigned			0	x + 0x50
12. alarm	(dyn.)	unsigned			0	x + 0x58
13. configuration	(dyn.)	long-unsigned			0	x + 0x60
14. encryption_key_status	(dyn.)	enum			0	x + 0x68
Méthodes spécifiques		***o/f***				
1. slave_install (data)		f				x + 0x70
2. slave_deinstall (data)		f				x + 0x78
3. capture (data)		f				x + 0x80
4. reset_alarm (data)		f				x + 0x88
5. synchronize_clock (data)		f				x + 0x90
6. data_send (data)		f				x + 0x98
7. set_encryption_key (data)		f				x + 0xA0
8. transfer_key (data)		f				x + 0xA8

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.3.2 Description d'attribut

4.8.3.2.1 logical_name

Identifie l'instance de l'objet "M-Bus client". Voir 6.2.22.

4.8.3.2.2 mbus_port_reference

Fournit une référence à un objet "M-Bus master port setup", utilisé pour configurer un port M-Bus, chaque interface permettant d'échanger des données avec un ou plusieurs dispositifs M-Bus esclaves.

4.8.3.2.3 capture_definition

Fournit la *capture_definition* pour les dispositifs M-Bus esclaves.

NOTE 1 Cet attribut peut être préalablement configuré ou écrit dans le cadre de la procédure d'installation.

```
array        capture_definition_element

capture_definition_element::= structure
{
data_information_block:      octet-string,
value_information_block:     octet-string
}
```

NOTE 2 Les éléments data_information_block et value_information_block correspondent au bloc d'informations de données (DIB) et au bloc d'informations sur les valeurs (VIB) respectivement décrits dans l'EN 13757-3:2013, 6.2 et Article 7.

4.8.3.2.4 capture_period

>= 1: Capture automatique prise par hypothèse. Spécifie la période de capture en secondes.

0: Aucune capture automatique: la capture est déclenchée en externe ou des événements de capture se produisent de manière asynchrone.

4.8.3.2.5 primary_address

Contient l'adresse primaire du dispositif M-Bus esclave. La plage est 0...250.

Chaque dispositif M-bus est lié à une voie du M-Bus maître. Toutefois, il n'existe aucun lien direct entre l'adresse primaire et le numéro de voie.

NOTE 1 La spécification du champ B des codes OBIS limite la plage de 1...64 à l'intérieur d'un dispositif logique. Voir IEC 62056-6-1:2023, 5.2.

Si le dispositif esclave est déjà configuré et donc que son adresse primaire est différente de 0, cette valeur doit être écrite dans l'attribut *primary_address*. À partir de cet instant, l'échange de données avec le dispositif de M-Bus esclave est possible.

Autrement, la méthode *slave_install* doit être utilisée (voir 4.8.3.3.1).

NOTE 2 L'attribut *primary_address* ne peut pas être utilisé pour stocker une adresse primaire souhaitée pour un dispositif esclave non configuré. Lorsque l'attribut *primary_address* est défini, cela signifie que le client M-Bus peut opérer immédiatement avec cette adresse primaire, ce qui n'est pas le cas avec un dispositif esclave non configuré.

4.8.3.2.6 identification_number

Contient l'élément "Identification Number" (numéro d'identification) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.5.

Cet attribut, ainsi que les attributs 7, 8 et 9, contiennent les valeurs trouvées dans le premier message reçu après l'installation.

Si ces valeurs sont différentes dans les messages suivants, le message est ignoré.

4.8.3.2.7 manufacturer_id

Contient l'élément "Manufacturer Identification" (identification du fabricant) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.6.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.3.2.8 version

Contient l'élément "Version" de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.7.

4.8.3.2.9 device_type

Contient l'élément "Device type identification" (identification du type de dispositif) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.8, Tableau 6.

4.8.3.2.10 access_number

Contient l'élément "Access Number" (numéro d'accès) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.9.

4.8.3.2.11 status

Contient l'élément "Status byte" (octet d'état) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.10, Tableaux 7 et 8.

Il est mis à jour à chaque lecture du dispositif M-Bus esclave.

4.8.3.2.12 alarm

Contient l'état "Alarm" (Alarme) spécifié dans l'EN 13757-3:2013, Annexe D.

Il est mis à jour à chaque lecture du dispositif M-Bus esclave.

4.8.3.2.13 configuration

Contient le champ Configuration (précédemment: champ Signature) tel que spécifié dans l'EN 13757-3:2013, 5.12. Il contient des informations sur le mode de cryptage et le nombre d'octets cryptés.

Il est mis à jour à chaque lecture du dispositif M-Bus esclave.

4.8.3.2.14 encryption_key_status

Donne des informations sur l'état de la clé de cryptage. Voir aussi Annexe B.

enum:

(0) pas de clé de cryptage;
(1) encryption_key défini;
(2) encryption_key transféré;
(3) encryption_key défini et transféré;
(4) encryption_key en cours d'utilisation.

4.8.3.3 Description de la méthode

4.8.3.3.1 slave_install (data)

Installe un dispositif esclave, qui est encore non configuré (son adresse primaire est 0).

data::= unsigned

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Cette méthode ne peut être appelée avec succès que si la valeur en cours de l'attribut *primary_address* est 0. Les actions suivantes sont entreprises:

– l'adresse M-Bus 0 est vérifiée pour détecter la présence d'un nouveau dispositif;
– si aucun M-Bus esclave non installé n'est trouvé, l'appel de la méthode échoue;
– si la méthode *slave_install* est appelée avec "0" comme paramètre, l'adresse primaire est automatiquement attribuée. Pour ce faire, l'attribut *primary_address* de tous les objets "M-Bus client" est vérifié dans le dispositif DLMS®/COSEM et en sélectionnant ensuite le premier numéro non utilisé. L'attribut *primary_address* est mis à cette adresse et est ensuite transféré au dispositif M-Bus esclave;
– si la méthode *slave_install* est appelée avec une adresse primaire (différente de 0) comme paramètre, l'attribut *primary_address* est défini sur cette valeur, puis transféré au dispositif M-Bus esclave.

NOTE Les dispositifs esclaves non configurés sont configurés avec l'adresse primaire telle que spécifiée dans l'EN 13757-3:2013, Annexe E.5.

4.8.3.3.2 slave_deinstall (data)

Désinstalle le dispositif esclave. Le but principal de ce service est de désinstaller le dispositif M-Bus esclave et de préparer le maître pour l'installation d'un nouveau dispositif. Les actions suivantes sont entreprises:

– l'adresse M-Bus est mise à 0 dans le dispositif M-Bus esclave;
– la clé de cryptage transférée précédemment au dispositif M-Bus esclave est détruite; la clé par défaut n'est pas altérée;
– *encryption_key_status* est défini sur (0): pas de clé de cryptage;
– l'attribut *primary_address* est également mis à 0.

NOTE Un nouvel esclave M-Bus ne peut être installé qu'une fois que la valeur de l'attribut *primary_address* est 0.

data::= unsigned (0)

4.8.3.3.3 capture (data)

Capture des valeurs (telles que spécifiées par l'attribut *capture_definition*) provenant du dispositif M-Bus esclave.

data::= integer (0)

4.8.3.3.4 reset_alarm (data)

Réinitialise l'état d'alarme du dispositif M-Bus esclave.

data::= integer (0)

4.8.3.3.5 synchronize_clock (data)

Synchronise l'horloge du dispositif M-Bus esclave avec celle du dispositif M-Bus client.

data::= integer (0)

4.8.4 Wireless Mode Q channel (class_id = 73, version = 1)

4.8.4.1 Vue d'ensemble

Les instances de cette IC définissent les paramètres opérationnels pour la communication en utilisant les interfaces de mode Q. Voir aussi EN 13757-5:2015.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Wireless Mode Q channel		0...n	class_id = 73, version = 1			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. addr_state	(static)	enum				x + 0x08
3. device_address	(static)	octet-string				x + 0x10
4. address_mask	(static)	octet-string				x + 0x18
Méthodes spécifiques		***o/f***				

4.8.4.2 Description d'attribut

4.8.4.2.1 logical_name

Identifie l'instance de l'objet "Wireless Mode Q channel". Voir 6.2.22.

4.8.4.2.2 addr_state

Définit si une adresse a été ou non attribuée au dispositif depuis la dernière mise sous tension du dispositif.

enum:

(0) n'a pas encore eu d'adresse attribuée;
(1) a une adresse attribuée, soit par réglage manuel, soit par une méthode automatisée.

4.8.4.2.3 device_address

L'adresse actuellement attribuée du dispositif sur le réseau.

4.8.4.2.4 address_mask

L'adresse de groupe à laquelle le dispositif répondra lorsque l'adressage de forme courte est utilisé.

4.8.5 M-Bus master port setup (class_id = 74, version = 0)

4.8.5.1 Vue d'ensemble

Les instances de cette IC définissent les paramètres opérationnels pour la communication en utilisant les interfaces de l'EN 13757-2 si le dispositif agit comme un M-Bus maître.

M-Bus master port setup		0...n	class_id = 74, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. comm_speed	(static)	enum	0	7	3	x + 0x08
Méthodes spécifiques		***o/f***				

4.8.5.2 Description d'attribut

4.8.5.2.1 logical_name

Identifie l'instance de l'objet "M-Bus master port setup". Voir 6.2.22.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.5.2.2 comm_speed

La vitesse de communication prise en charge par le port.

enum:

(0) 300 bauds;
(1) 600 bauds;
(2) 1 200 bauds;
(3) 2 400 bauds;
(4) 4 800 bauds;
(5) 9 600 bauds;
(6) 19 200 bauds;
(7) 38 400 bauds.

4.8.6 DLMS®/COSEM server M-Bus port setup (class_id = 76, version = 0)

4.8.6.1 Vue d'ensemble

Les instances de l'IC "DLMS®/COSEM server M-Bus port setup" sont utilisées dans les serveurs DLMS®/COSEM hébergés par des dispositifs M-Bus esclaves, à l'aide d'un profil de communication M-Bus filaire et sans fil DLMS®/COSEM (wM-Bus).

DLMS/COSEM server M-Bus port setup		**0...n**	**class_id = 76, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. M-Bus_profile_selection	(static)	octet-string				x + 0x08
3. M-Bus_port_communication_state	(dyn.)	enum				x + 0x10
4. M-Bus_Data_Header_Type	(dyn.)	enum			2	x + 0x18
5. primary_address	(static)	unsigned			0	x + 0x20
6. identification_number	(static)	double-long-unsigned			0	x + 0x28
7. manufacturer_id	(static)	long-unsigned			0	x + 0x30
8. version	(static)	unsigned			0	x + 0x38
9. device_type	(static)	unsigned			0	x + 0x40
10. max_pdu_size	(static)	long-unsigned				x + 0x48
11. listening_window	(static)	array				x + 0x50
Méthodes spécifiques		***o/f***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.6.2 Description d'attribut

4.8.6.2.1 logical_name

Identifie l'instance de l'objet "DLMS®/COSEM server M-Bus port setup". Voir 6.2.22.

4.8.6.2.2 M-Bus_profile_selection

Fait référence à un objet de définition de port de communication M-Bus décrivant les capacités physiques pour une communication filaire et sans fil. L'objet référencé est soit un objet "M-Bus slave port setup" (class_id = 25) soit un objet "Wireless Mode Q channel" (class_id = 73).

4.8.6.2.3 M-Bus_port_communication_state

Cet attribut est uniquement pertinent dans wM-Bus.

Contient l'état de communication du nœud M-Bus.

Voir EN 13757-4:2013, 11.6.3, Tableau 27.

enum:

(0) aucun accès: le compteur ne fournit aucune fenêtre d'accès (compteur unidirectionnel);

(1) provisoirement aucun accès: le compteur prend en charge l'accès bidirectionnel en général, mais il n'y a aucune fenêtre d'accès après cette transmission (provisoirement aucun d'accès afin de rester dans les limites du cycle de service ou pour limiter la consommation d'énergie);

(2) accès limité: le compteur fournit uniquement une courte fenêtre d'accès immédiatement après cette transmission (compteur fonctionnant sur batterie, par exemple);

(3) accès illimité: le compteur fournit un accès illimité jusqu'à la prochaine transmission au moins (dispositifs alimentés sur secteur, par exemple).

4.8.6.2.4 M-Bus_Data _Header_Type

Contient le type de l'en-tête de données M-Bus, déduit de la valeur CITL issue de la communication en cours.

Dans le cas d'une opération Push, la valeur par défaut (2) doit être appliquée.

enum:

(0) M-Bus_Data_Header_Type == None_M-Bus_Data_Header, la valeur du champ CITL doit être 0x10 lorsque la segmentation n'est pas utilisée, et doit être 0x0 .. 0x1F lorsque la segmentation est utilisée (avec le bit FIN défini sur 0 dans tous les segments, sauf le dernier);

(1) M-Bus_Data_Header_Type == Short_M-Bus_Data_Header, la valeur du champ CITL doit être 0x61 dans une trame M-Bus envoyée par un maître et 0x7D dans une trame M-Bus envoyée par un esclave;

(2) M-Bus_Data_Header_Type == Long_M-Bus_Data_Header, la valeur du champ CITL doit être 0x60 dans une trame M-Bus envoyée par un maître et 0x7C dans une trame M-Bus envoyée par un esclave;

4.8.6.2.5 primary_address

Contient l'adresse primaire du dispositif M-Bus esclave.

Voir IEC 62056-7-3:2017, Tableau 1.

Si le dispositif esclave est déjà configuré et donc que son adresse primaire est différente de 0, l'échange de données avec le dispositif M-Bus esclave est possible.

4.8.6.2.6 identification_number

Contient l'élément "Identification Number" (numéro d'identification) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.5.

Dans le cas de la communication unidirectionnelle, cet attribut (avec les attributs 6, 7, 8 et 9) doit être paramétré.

Dans le cas de la communication bidirectionnelle, cet attribut (avec les attributs 6, 7, 8 et 9) contient les valeurs trouvées dans le premier message reçu après l'installation par différentes interfaces de gestion de réseau M-Bus.

NOTE Si ces valeurs sont différentes dans les messages suivants, le message est ignoré.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.6.2.7 manufacturer_id

Contient l'élément "Manufacturer Identification" (identification du fabricant) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.6.

4.8.6.2.8 version

Contient l'élément "Version" de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.7.

4.8.6.2.9 device_type

Contient l'élément "Device type identification" (identification du type de dispositif) de l'en-tête de données tel que spécifié dans l'EN 13757-3:2013, 5.8, Tableau 6.

4.8.6.2.10 max_pdu_size

Contient la capacité de longueur disponible provenant des couches basses M-Bus (en octets).

Spécifie la longueur maximale des données utiles DLMS® (une APDU ou l'une de ses parties) qu'une trame M-Bus peut contenir.

Dans le cas des messages longs, le transfert par bloc assuré par la couche application DLMS®/COSEM et/ou la segmentation assurée par la couche transport peuvent être utilisés.

4.8.6.2.11 listening_window

Cet attribut est uniquement pertinent dans wM-Bus.

Contient les points dans le temps où la fenêtre de communication devient active (start_time) et inactive (end_time). start_time définit implicitement la période.

EXEMPLE Lorsque le jour du mois n'est pas spécifié (égal à 0xFF), cela signifie qu'il s'agit de la gestion d'une fenêtre d'écoute quotidienne. Une gestion de fenêtre quotidienne, mensuelle, etc., peut être définie.

```
array window_element

window_element::= structure

{
start_time: octet-string,
end_time: octet-string
}
```

start_time et end_time sont formatés comme spécifié en 4.6.1 pour *date-time*.

4.8.7 M-Bus diagnostic (class_id = 77, version = 0)

4.8.7.1 Vue d'ensemble

Les instances de l'IC "M-Bus diagnostic" contiennent les informations relatives au fonctionnement du réseau M-Bus (l'intensité du signal de courant, l'identifiant de voie, l'état de la liaison avec le réseau M-Bus et les compteurs liés à l'échange de trames, à la transmission et à la qualité de réception des trames, par exemple).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

M-Bus diagnostic		0...n	class_id = 77, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				X
2. received-signal-strength	(dyn.)	unsigned				x + 0x08
3. channel_Id	(dyn.)	unsigned			0	x + 0x10
4. link_status	(dyn.)	enum				x + 0x18
5. broadcast_frames_counter	(dyn.)	array			0	x + 0x20
6. transmissions_counter	(dyn.)	double-long-unsigned			0	x + 0x28
7. FCS_OK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x30
8. FCS_NOK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x38
9. capture_time	(dyn.)	structure				x + 0x40
Méthodes spécifiques		***o/f***				
1. reset (data)		f				x + 0x48

4.8.7.2 Description d'attribut

4.8.7.2.1 logical_name

Identifie l'instance de l'objet "M-Bus diagnostic". Voir 6.2.22.

4.8.7.2.2 received_signal_strength

Cet attribut est uniquement pertinent pour la communication M-Bus bidirectionnelle sans fil.

Si le profil wM-Bus est utilisé, cet attribut contient la valeur de l'intensité du signal de la dernière trame wM-Bus reçue, en dBm.

4.8.7.2.3 channel_Id

Cet attribut est uniquement pertinent pour la communication wM-Bus.

Si le profil wM-Bus est utilisé, cet attribut contient l'identification de la voie actuellement utilisée.

La valeur par défaut est 0.

4.8.7.2.4 link_status

Cet attribut est uniquement pertinent pour la communication wM-Bus.

Si le profil wM-Bus est utilisé, cet attribut contient l'état en cours de la liaison avec le réseau M-Bus.

Les états possibles sont spécifiés dans l'EN 13757-5:2015, 9.7.4.1.7.

enum:

(0) par défaut (données jamais reçues);
(1) liaison en fonctionnement normal;
(2) liaison temporairement interrompue;
(3) liaison définitivement interrompue.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE En cas de perte de synchronisation avec le réseau, les tentatives de rétablissement de la liaison sont automatiques en procédant à des analyses radio. Tant que les tentatives de resynchronisation sont en cours, link_status est temporairement interrompu.

Link_status passe en fonctionnement normal lorsque la liaison est rétablie.

Link_status est définitivement interrompu lorsque le nombre de tentatives spécifié par le fabricant est dépassé.

4.8.7.2.5 broadcast_frames_counter

Contient les valeurs "broadcast-frames-counter" avec l'horodatage de la dernière trame reçue et se distinguant par les identifiants de client.

```
array        broadcast_frame_counter_definition

broadcast_frame_counter_definition::=    structure

{
      client_id:        unsigned,
      counter:          double-long-unsigned,
      time_stamp:       date-time
}
```

La valeur par défaut est 0.

4.8.7.2.6 transmissions_counter

Compte le nombre de trames transmises par le port M-Bus associé.

Le compteur d'émissions est incrémenté au début de chaque phase d'émission. Un système client peut écrire cette variable pour mettre à jour le compteur. Lorsque le compteur d'émissions atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément.

La valeur par défaut est 0.

4.8.7.2.7 FCS_OK_frames_counter

Compte le nombre de trames reçues avec une somme de contrôle correcte. Lorsque le champ FCS_OK_frames_counter atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément.

La valeur par défaut est 0.

4.8.7.2.8 FCS_NOK_frames_counter

Compte le nombre de trames reçues avec une somme de contrôle incorrecte. Lorsque le champ FCS_NOK frames counter atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément. La valeur par défaut est 0.

4.8.7.2.9 capture_time

Contient l'horodatage du dernier changement de valeur de l'attribut 2, 4, 6, 7 ou 8.

```
capture_time::=    structure

{
      attribute_id:     unsigned,
      time_stamp:       date-time
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.8.7.3 Description de la méthode

4.8.7.3.1 reset (data)

Efface tous les counters, received_signal_strength, link_status et capture_time.

data::= integer(0)

NOTE La gestion de channel_id est hors du domaine d'application de la spécification de cette IC.

Voir EN 13757-4:2013.

4.9 Classes d'interfaces pour l'établissement d'échange de données sur Internet

4.9.1 TCP-UDP setup (class_id = 41, version = 0)

4.9.1.1 Vue d'ensemble

Cette IC permet de modéliser l'établissement de la sous-couche TCP ou UDP de la couche transport COSEM basée sur TCP ou UDP d'un profil de communication basé sur TCP-UDP/IP.

Dans les profils de communication basés sur TCP-UDP/IP, toutes les AA entre un dispositif physique hébergeant un ou plusieurs processus d'application client COSEM et un dispositif physique hébergeant un ou plusieurs AP serveur COSEM s'appuient sur une seule connexion TCP ou UDP. L'entité TCP ou UDP est enveloppée dans la couche transport COSEM basée sur TCP-UDP. Au sein d'un dispositif physique, chaque AP (AP client ou dispositif logique serveur) est lié à un port conteneur (WPort). La liaison se fait à l'aide de l'objet "SAP Assignment" (voir 4.4.5).

D'autre part, une couche transport COSEM basée sur TCP ou sur UDP peut être capable de prendre en charge plusieurs connexions TCP ou UDP, entre un dispositif physique et plusieurs dispositifs physiques homologues hébergeant des AP COSEM.

Lorsqu'un dispositif physique COSEM prend en charge diverses couches liaison de données (Ethernet et PPP, par exemple), une instance de l'objet "TCP-UDP setup" est nécessaire pour chacune d'elles.

TCP-UDP setup		**0...n**	**class_id = 41, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. TCP-UDP_port	(static)	long-unsigned				x + 0x08
3. IP_reference	(static)	octet-string				x + 0x10
4. MSS	(static)	long-unsigned	40	65...535	576	x + 0x18
5. nb_of_sim_conn	(static)	unsigned	1			x + 0x20
6. inactivity_time_out	(static)	long-unsigned			180	x + 0x28
Méthodes spécifiques		***o/f***				

4.9.1.2 Description d'attribut

4.9.1.2.1 logical_name

Identifie l'instance de l'objet "TCP-UDP setup". Voir 6.2.23.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.1.2.2 TCP-UDP_port

Contient le numéro d'accès TCP-UDP sur lequel le dispositif physique est à l'écoute pour détecter l'application DLMS®/COSEM.

Pour DLMS®/COSEM, les numéros d'accès suivants ont été enregistrés par l'IANA. Voir http://www.iana.org/assignments/port-numbers.

dlms/cosem 4059/TCP DLMS/COSEM

dlms/cosem 4059/UDP DLMS/COSEM

4.9.1.2.3 IP_reference

Référence un objet "IP setup" par son nom logique. L'objet référencé contient des informations relatives aux réglages de l'adresse IP de la couche IP prenant en charge la couche TCP-UDP.

4.9.1.2.4 MSS

Avec l'aide de l'option MSS (taille de segment maximale), une couche TCP peut indiquer à son partenaire la taille maximale de segment de réception. A noter que:

– cette option doit seulement être envoyée dans la demande de connexion initiale (c'est-à-dire dans des segments avec le bit de commande SYN envoyé);
– par convention, si cette option est absente, la valeur par défaut (576) de la MSS est prise en compte;
– la MSS n'est pas négociable; sa valeur est indiquée par son attribut.

4.9.1.2.5 nb_of_sim_conn

Le nombre maximal de connexions simultanées que la couche transport COSEM basée sur TCP-UDP est capable de prendre en charge.

4.9.1.2.6 inactivity_time_out

Définit le temps, en secondes, au-delà duquel, si aucune trame n'est reçue en provenance du client COSEM, la connexion TCP inactive doit être abandonnée.

Lorsque cette valeur est mise à 0, cela signifie que *inactivity_time_out* n'est pas opérationnel. En d'autres termes, une connexion TCP, une fois établie, dans des conditions normales (pas de coupure secteur, etc.) ne sera jamais abandonnée par le serveur COSEM.

À noter que toutes les actions liées à la gestion de la fonction de dépassement de délai d'inactivité (mesure du temps d'inactivité, abandon de la connexion TCP si la temporisation est terminée, etc.) sont gérées dans la mise en œuvre de la couche TCP-UDP.

4.9.2 IPv4 setup (class_id = 42, version = 0)

4.9.2.1 Vue d'ensemble

NOTE Comparée aux éditions antérieures du présent document, la présente spécification apporte des améliorations dans la présentation des attributs. Ne s'agissant pas de modifications techniques, la version de l'IC reste 0.

Cette IC permet de modéliser l'établissement de la couche IPv4, de gérer toutes les informations liées aux réglages d'adresses IP associés à un dispositif donné et à une connexion de couche basse sur laquelle ces réglages sont utilisés.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Il doit y avoir une instance de cette IC dans un dispositif pour chaque différente interface réseau mise en œuvre. Par exemple, pour un dispositif à deux interfaces (utilisant le profil TCP-UDP/IPv4 sur les deux), il doit y avoir deux instances de l'IC "IPv4 setup" dans le dispositif en question: une pour chacune de ces interfaces.

IPv4 setup	**0...n**	**class_id = 42, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. DL_reference (static)	octet-string				x + 0x08
3. IP_address	double-long-unsigned				x + 0x10
4. multicast_IP_address	array				x + 0x18
5. IP_options	array				x + 0x20
6. subnet_mask	double-long-unsigned				x + 0x28
7. gateway_IP_address	double-long-unsigned				x + 0x30
8. use_DHCP_flag (static)	boolean				x + 0x38
9. primary_DNS_address	double-long-unsigned				x + 0x40
10. secondary_DNS_address	double-long-unsigned				x + 0x48
Méthodes spécifiques	***o/f***				
1. add_mc_IP_address (data)	f				x + 0x60
2. delete_mc_IP_address (data)	f				x + 0x68
3. get_nbof_mc_IP_addresses (data)	f				x + 0x70

4.9.2.2 Description d'attribut

4.9.2.2.1 logical_name

Identifie l'instance de l'objet "IPv4 setup". Voir 6.2.23.

4.9.2.2.2 DL_reference

Référence un objet d'établissement de couche liaison de données (Ethernet ou PPP, par exemple) par son nom logique.

L'objet référencé contient des informations relatives aux réglages spécifiques de la couche liaison de données prenant en charge la couche IP.

4.9.2.2.3 IP_address

Contient la valeur de l'adresse IP (IPv4) de ce dispositif physique sur le réseau auquel le dispositif est connecté via l'interface de couche inférieure référencée. Il peut être "static" (statique) ou "dynamic" (dynamique). Dans le second cas, l'affectation d'adresse IP dynamique (DHCP, par exemple) est utilisée.

Si aucune adresse IP n'est assignée, la valeur est 0.

EXEMPLE L'adresse IPv4 192.168.0.1 (en notation décimale à point) correspond à C0A80001 (hexa), ce qui donne 3232235521 (double-long-unsigned).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.2.2.4 multicast_IP_address

Contient un tableau d'adresses IP. Les adresses IP dans ce tableau doivent s'inscrire dans la plage d'adresses de groupe de multidiffusion (adresses de "classe D", y compris les adresses IP dans la plage de 224.0.0.0 à 239.255.255.255). Lorsqu'un dispositif reçoit un datagramme IP avec l'une de ces adresses IP dans le champ d'adresse IP de destination, il doit considérer que ce datagramme lui est adressé.

multicast_IP_address::= array double-long-unsigned

4.9.2.2.5 IP_options

Contient les paramètres nécessaires à la prise en charge des options IP sélectionnées – par exemple horodatage de datagrammes ou services de sécurité (IPSec).

```
IP_options::= array      IP_options_element

IP_options_element::= structure

{
IP_Option_Type:          unsigned,
IP_Option_Length:        unsigned,
IP_Option_Data:          octet-string
}
```

NOTE Dans tous les cas, comme cela est spécifié dans la RFC 791, le champ IP_Option_Length inclut la longueur totale des trois champs: IP_Option_Type, IP_Option_Length and IP_Option_Data.

IP_Option_Types admis:

a) Security: IP_Option_Type = 0x82, IP_Option_Length = 11

Si cette option est présente, le dispositif doit être autorisé à envoyer des paramètres de sécurité, de cloisonnement, de restrictions de gestion et de TCC (groupe d'utilisateurs fermé) dans ses datagrammes IP. IP_Option_Data doit contenir la valeur des données de sécurité, de cloisonnement, de gestion des restrictions et le code de pays de contrôle de la transmission, tels que spécifiés dans la RFC 791.

b) Loose Source et Record Route: IP_Option_Type = 0x83

Si cette option est présente, le dispositif doit fournir des informations d'acheminement devant être utilisées par les passerelles pour transmettre le datagramme à la destination et pour enregistrer les informations d'acheminement. Les valeurs d'IP_Option_Length et d'IP_Option_Data sont spécifiées dans la RFC 791.

c) Strict Source et Record Route: IP_Option_Type = 0x89

Si cette option est présente, le dispositif doit fournir des informations d'acheminement devant être utilisées par les passerelles pour transmettre le datagramme à la destination et pour enregistrer les informations d'acheminement. Les valeurs d'IP_Option_Length et d'IP_Option_Data sont spécifiées dans la RFC 791.

d) Record Route: IP_Option_Type = 0x07

Si cette option est présente, le dispositif doit en plus:

– envoyer des datagrammes IP émis avec cette option, en fournissant des moyens d'enregistrer le chemin de ces datagrammes;

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– en tant que routeur, envoyer des datagrammes IP acheminés avec l'option d'acheminement réglée en fonction de cette option.

Les valeurs d'IP_Option_Length et d'IP_Option_Data sont spécifiées dans la RFC 791.

e) Internet Timestamp: IP_Option_Type = 0x44

Si cette option est présente, le dispositif doit en plus:

– envoyer des datagrammes IP émis avec cette option, en fournissant des moyens pour horodater le datagramme dans le chemin vers sa destination;
– en tant que routeur, envoyer des datagrammes IP acheminés avec l'option d'horodatage réglée en fonction de cette option.

Les valeurs d'IP_Option_Length et d'IP_Option_Data sont spécifiées dans la RFC 791.

4.9.2.2.6 subnet_mask

Contient le masque de sous-réseau.

Lorsque le sous-réseautage est utilisé dans un segment de réseau, chaque dispositif concerné doit se comporter conformément aux règles de sous-réseautage. Pour ce faire, le dispositif, outre son adresse IP, a besoin de connaître également la structure de l'adresse IP au sein du segment mis en sous-réseau. L'attribut subnet_mask contient cette information.

Avec IPv4, le *subnet_mask* est un mot de 32 bits, exprimé exactement dans le même format qu'une adresse IP (255.255.255.0, par exemple), mais a une autre signification: les bits "0" du *subnet_mask* indiquent la partie de l'adresse IP qui est encore utilisée comme Device_ID sur un réseau IP mis en sous-réseau[3].

4.9.2.2.7 gateway_IP_address

Contient l'adresse IP du dispositif passerelle.

Dans la plupart des mises en œuvre IP, il existe un code dans le module qui gère les datagrammes sortants pour décider si un datagramme peut être envoyé directement à la destination sur le réseau local ou s'il doit être envoyé vers une passerelle. Afin de pouvoir envoyer des datagrammes non locaux vers la passerelle, le dispositif doit connaître l'adresse IP du dispositif passerelle assignée au segment de réseau donné.

Si aucune adresse IP n'est assignée, la valeur est 0.

4.9.2.2.8 use_DHCP_flag

TRUE: Le dispositif utilise le protocole DHCP (protocole de configuration d'hôte dynamique) pour déterminer dynamiquement les paramètres *IP_address*, *subnet_mask* et *gateway_IP_address*.

FALSE: Les paramètres *IP_address, subnet_mask* et *gateway_IP_address* doivent être définis localement.

4.9.2.2.9 primary_DNS_address

L'adresse IP du serveur de noms de domaine (DNS) primaire.

Si aucune adresse IP n'est assignée, la valeur est 0.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.2.2.10 secondary_DNS_ address

L'adresse IP du serveur de noms de domaine (DNS) secondaire.

Si aucune adresse IP n'est assignée, la valeur est 0.

4.9.2.3 Description de la méthode

4.9.2.3.1 add_mc_IP_address (IP_Address)

Ajoute une adresse IP de multidiffusion au tableau *multicast_IP_address*.

IP_Address::= double-long-unsigned

4.9.2.3.2 delete_mc_IP_address (IP_Address)

Supprime une adresse IP du tableau *multicast_IP_Address*. L'adresse IP à effacer est identifiée par sa valeur.

IP_Address::= double-long-unsigned

4.9.2.3.3 get_nbof_mc_IP_addresses (data)

Retourne le nombre d'adresses IP contenues dans le tableau *multicast_IP_Address*.

data::= unsigned

4.9.3 IPv6 setup (class_id = 48, version = 0)

4.9.3.1 Vue d'ensemble

NOTE Voir aussi Annexe C.

L'IC "IPv6 setup" permet de modéliser l'établissement de la couche IPv6, de gérer toutes les informations liées aux réglages d'adresses IPv6 associés à un dispositif donné et à une connexion de couche inférieure sur laquelle ces réglages sont utilisés.

Il doit y avoir une instance de cette IC dans un dispositif pour chaque différente interface réseau mise en œuvre. Par exemple, si un dispositif a deux interfaces (utilisant le profil UDP/IP et/ou TCP/IP sur les deux), il doit y avoir deux instances de l'IC "IPv6 setup" dans le dispositif en question: une pour chacune de ces interfaces.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

IPv6 setup		0...n	class_id = 48, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. DL_reference	(static)	octet-string				x + 0x08
3. address_config_mode	(static)	enum			0	x + 0x10
4. unicast_IPv6_addresses		array				x + 0x18
5. multicast_IPv6_addresses	(static)	array				x + 0x20
6. gateway_IPv6_addresses	(static)	array			0	x + 0x28
7. primary_DNS_address	(static)	octet-string			0	x + 0x30
8. secondary_DNS_address	(static)	octet-string			0	x + 0x38
9. traffic_class	(static)	unsigned	0	63	0	x + 0x40
10. neighbor_discovery_setup	(static)	array				x + 0x48
Méthodes spécifiques		***o/f***				
1. add_IPv6_address (data)		*f*				x + 0x60
2. remove_IPv6_address (data)		*f*				x + 0x68

4.9.3.2 Description d'attribut

4.9.3.2.1 logical_name

Identifie l'instance de l'objet "IPv6 setup". Voir 6.2.23.

4.9.3.2.2 DL_reference

Référence un objet "Data link layer setup" par son nom logique. L'objet référencé contient des informations relatives aux réglages spécifiques de la couche liaison de données prenant en charge la couche IPv6.

4.9.3.2.3 address_config_mode

Définit le mode de configuration de l'adresse IPv6.

enum:

(0) Auto-configuration (par défaut);
(1) DHCPv6;
(2) Manuel;
(3) ND (découverte de voisins).

NOTE L'address_config_mode est commun à toutes les adresses IPv6 gérées par une instance de la classe "IPv6 setup".

4.9.3.2.4 unicast_IPv6_addresses

Contient les adresses IPv6 unicast attribuées à l'interface concernée du dispositif physique sur le réseau (Unique Local Unicast, Link Local Unicast et/ou Global Unicast). Une adresse IPv6 peut être "static" (statique) et/ou "dynamic" (dynamique).

unicast_IPv6_addresses::= array octet-string

Le format de chaque adresse IPv6 unicast doit être tel que spécifié dans la RFC 3513.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Si aucune adresse IPv6 unicast n'est attribuée à l'interface, un tableau ne contenant aucun élément est présent (par défaut).

Pour réinitialiser toutes les adresses IPv6 unicast configurées, un tableau ne contenant aucun élément doit être écrit.

4.9.3.2.5 multicast_IPv6_addresses

Contient un tableau d'adresses IPv6 utilisées pour la multidiffusion.

multicast_IPv6_addresses::= array octet-string

Le format de chaque adresse IPv6 multicast doit être tel que spécifié dans la RFC 3513.

Si aucune adresse IPv6 multicast n'est attribuée à l'interface, un tableau ne contenant aucun élément est présent (par défaut).

Pour réinitialiser toutes les adresses IPv6 multicast configurées, un tableau ne contenant aucun élément doit être écrit.

4.9.3.2.6 gateway_IPv6_addresses

Contient les adresses IPv6 du dispositif de passerelle IPv6.

gateway_IPv6_addresses::= array octet-string

Le format de chaque adresse IPv6 de passerelle doit être tel que spécifié dans la RFC 3513.

Si aucune adresse IPv6 de passerelle n'est attribuée à l'interface, un tableau ne contenant aucun élément est présent (par défaut).

Pour réinitialiser toutes les adresses IPv6 de passerelle configurées, un tableau ne contenant aucun élément doit être écrit.

4.9.3.2.7 primary_DNS_address

Contient l'adresse IPv6 du serveur de noms de domaine (DNS) primaire.

Si aucune adresse IPv6 n'est attribuée, la longueur de l'octet-string doit être 0.

4.9.3.2.8 secondary_DNS_address

Contient l'adresse IPv6 du serveur de noms de domaine (DNS) secondaire.

Si aucune adresse IPv6 n'est attribuée, la longueur de l'octet-string doit être 0.

4.9.3.2.9 traffic_class

Contient l'élément de classe de trafic de l'en-tête IPv6. Les 6 bits de poids fort sont utilisés pour le code DSCP (code d'accès aux services différenciés), qui est utilisé pour classer les paquets conformément à la RFC 2474:1998, Article 3.

4.9.3.2.10 neighbor_discovery_setup

Contient la configuration à utiliser pour les routeurs et les hôtes pour prendre en charge le protocole ND (découverte des voisins) pour IPv6 (RFC 4861).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
array neighbor_discovery_setup

neighbor_discovery_setup::= structure

{
    RS_max_retry:          unsigned,
    RS_retry_wait_time:    long-unsigned,
    RA_send_period:        double-long-unsigned
}
```

Où:

RS_max_retry — Donne le nombre maximal de nouvelles tentatives de sollicitation de routeur à effectuer par un nœud si l'annonce de routeur prévue n'a pas été reçue.
Plage: 1-255
Valeur par défaut: 3

RS_retry_wait_time — Donne le temps d'attente en millisecondes entre deux nouvelles tentatives successives de sollicitation de routeur.
Plage: 0-65 535
Valeur par défaut: 10 000

RA_send_period — Donne la période de transmission de l'annonce de routeur, en secondes.
Plage: 0-4 294 967 295

4.9.3.3 Description de la méthode

4.9.3.3.1 add_IPv6_address (data)

Ajoute une adresse IPv6 de l'interface physique au tableau d'adresses IPv6.

```
data::= structure
{
    IPv6_address_type:     enum,
    IPv6_address:          octet-string
}
```

où IPv6_address_type définit le type d'adresse IPv6 à ajouter:

```
enum:

(0)  unicast;
(1)  multicast;
(2)  passerelle.
```

IPv6_address spécifie l'adresse IPv6 à ajouter. La nouvelle adresse est automatiquement ajoutée à la fin de la liste. Si une adresse est à ajouter à une position spécifique, cela peut être fait en écrivant l'ensemble du tableau.

4.9.3.3.2 remove_IPv6_address (data)

Retire une adresse IPv6 de l'interface physique du tableau d'adresses IPv6.

```
data::= structure

{
IPv6_address_type:     enum,
IPv6_address:          octet-string
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

où IPv6_address_type définit le type d'adresse IPv6 à retirer:

enum:

(0) unicast;
(1) multicast;
(2) passerelle.

IPv6_address spécifie l'adresse IPv6 à retirer.

4.9.4 MAC address setup (class_id = 43, version = 0)

4.9.4.1 Vue d'ensemble

NOTE 1 Le nom et l'usage de cette classe d'interfaces ont été changés de "Ethernet setup" en "MAC address setup" pour permettre une utilisation plus générale, sans changer la version.

Les instances de cette IC contiennent l'adresse MAC du dispositif physique (ou, plus généralement, d'un dispositif ou d'un logiciel). Il doit y avoir une instance de cette IC pour chaque interface réseau d'un dispositif physique.

NOTE 2 Dans le cas du profil de communication basé sur HDLC à trois couches, l'adresse MAC (adresse HDLC inférieure) est contenue dans un objet "IEC HDLC setup".

NOTE 3 Dans le cas du profil de communication S-FSK PLC, l'adresse MAC est contenue dans un objet "S-FSK Phy&MAC setup".

MAC address setup	**0...n**	**class_id = 43, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. MAC_address	octet-string				x + 0x08
Méthodes spécifiques	***o/f***				

4.9.4.2 Description d'attribut

4.9.4.2.1 logical_name

Identifie l'instance de l'objet "MAC address setup". Voir 6.2.21, 6.2.23 et 6.2.27.

4.9.4.2.2 mac_address

Contient l'adresse MAC.

4.9.5 PPP setup (class_id = 44, version = 0)

4.9.5.1 Vue d'ensemble

NOTE Comparée aux éditions antérieures du présent document, la présente spécification apporte des améliorations dans la présentation des attributs. Ne s'agissant pas de modifications techniques, la version de l'IC reste 0.

Cette IC permet de modéliser l'établissement d'interfaces utilisant le protocole PPP, en gérant toutes les informations liées aux réglages de PPP associés à un dispositif physique donné et à une connexion de couche inférieure sur laquelle ces réglages sont utilisés. Il doit y avoir une instance de cette IC pour chaque interface réseau d'un dispositif physique.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PPP setup		0...n	class_id = 44, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. PHY_reference	(static)	octet-string				x + 0x08
3. LCP_options	(static)	LCP_options_type				x + 0x10
4. IPCP_options	(static)	IPCP_options_type				x + 0x18
5. PPP_authentication	(static)	PPP_auth_type				x + 0x20
Méthodes spécifiques		***o/f***				

4.9.5.2 Description d'attribut

4.9.5.2.1 logical_name

Identifie l'instance de l'objet "PPP setup". Voir 6.2.23.

4.9.5.2.2 PHY_reference

Référence un autre objet par son logical_name. L'objet référencé contient des informations relatives à l'interface de couche physique spécifique, prenant en charge la couche PPP.

4.9.5.2.3 LCP_options

Cet attribut contient les paramètres nécessaires à la prise en charge des options de configuration LCP sélectionnées.

```
LCP_options_type::= array LCP_options_type_element

LCP_options_type_element::=      structure

{
LCP_Option_Type:            unsigned,
LCP_Option_Length:          unsigned,
LCP_Option_Data:            CHOICE
{
        structure               [2] -- pour Callback-data
        boolean                 [3] -- pour ProtF-Compr et AdCtr-Compr,
        double-long-unsigned    [6] -- pour ACCM et Mag-Num,
        unsigned                [17] -- pour FCS-Alternatives,
        long-unsigned           [18] -- pour MRU et Auth-Prot
}
}
```

NOTE 1 Dans tous les cas, comme cela est spécifié dans l'IETF STD 51/RFC 1661, le champ LCP_Option_Length inclut la longueur totale des trois champs: LCP_Option_Type, LCP_Option_Length et LCP_Option_Data.

NOTE 2 Pour les valeurs attribuées, voir les affectations des champs du protocole point à point (PPP), à l'adresse suivante: http://www.iana.org/assignments/ppp-numbers/ppp-numbers.xml

Les LCP_Option_Types pris en charge sont les suivants:

a) Maximum-Receive-Unit (MRU), LCP_Option_Type = 1. Voir IETF STD 51 / RFC 1661.

Cette option de configuration peut être envoyée pour informer l'homologue que l'implémentation réalisée peut recevoir des paquets plus gros ou pour demander à l'homologue d'envoyer des paquets plus petits. La valeur par défaut est 1 500 octets.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

b) Async-Control-Character-Map (ACCM), LCP_Option_Type = 2. Voir IETF STD 51 / RFC 1662.

Cette option de configuration fournit une méthode pour négocier l'utilisation de la transparence des caractères de contrôle sur des liaisons asynchrones.

c) Authentication-Protocol, LCP_Option_Type = 3. Voir IETF STD 51 / RFC 1661.

Cette option de configuration fournit une méthode pour négocier l'utilisation d'un protocole spécifique pour l'authentification. Par défaut, l'authentification n'est pas exigée. La valeur indique le protocole d'authentification utilisé sur la liaison PPP donnée. Les valeurs possibles sont:

0x0000 – Aucun protocole d'authentification n'est utilisé;

0xc023 – Le protocole PAP est utilisé;

0xc223 – Le protocole CHAP est utilisé;

0xc227 – Le protocole EAP est utilisé.

d) Magic-Number, LCP_Option_Type = 5. Voir IETF STD 51 / RFC 1661.

Cette option de configuration fournit une méthode pour détecter des liaisons fonctionnant en boucle et autres anomalies de la couche liaison de données.

e) Protocol-Field-Compression (PFC), LCP_Option_Type = 7. Voir IETF STD 51 / RFC 1661.

Cette option de configuration fournit une méthode pour négocier la compression des champs de protocole PPP.

f) Address-and-Control-Field-Compression (ACFC), LCP_Option_Type = 8. Voir IETF STD 51 / RFC 1661.

Cette option de configuration fournit une méthode pour négocier la compression des champs adresse et commande de la couche liaison de données.

g) FCS-Alternatives, LCP_Option_Type = 9. Voir RFC 1570.

Cette option de configuration fournit une méthode de mise en œuvre pour spécifier un autre format de FCS à envoyer par l'homologue ou pour négocier la FCS également. La valeur du champ "FCS-Alter (FCS Alternatives) options" identifie la FCS utilisée. Ce champ contient un seul octet. Il est composé de la valeur "logique" ou des valeurs suivantes:

bit 1 FCS nul;

bit 2 CCITT 16-bit FCS;

bit 4 CCITT 32-bit FCS.

h) Callback, LCP_Option_Type = 13. Voir RFC 1570.

Cette option de configuration fournit une méthode de mise en œuvre pour demander à un homologue joint de retourner l'appel. Cela procure une sécurité renforcée en assurant que le site distant ne peut se connecter qu'à partir d'un seul emplacement tel que défini par le numéro de rappel.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
callback_data::= structure

{
callback_active:          boolean, // default: false,
callback_data_length:     unsigned,
callback_operation:       unsigned,
callback_message:         octet-string
}
```

Où:

– le champ callback-active indique si l'option de rappel est active sur cette liaison PPP;

– le champ callback_operation indique le contenu du champ Message;

callback_operation: unsigned

(0) Emplacement déterminé par l'authentification d'utilisateur,
(1) Chaîne de numérotation,
(2) Identifiant d'emplacement,
(3) Numéro E.164,
(4) Nom distinctif X.500,
(5) Non attribué,
(6) L'emplacement est déterminé lors de la négociation CBCP.

le champ callback_message est zéro ou plus d'octets, et son contenu général est déterminé par le champ callback_operation. Le format réel de l'information est spécifique au site ou à l'application.

4.9.5.2.4 IPCP_options

Contient les paramètres nécessaires pour le protocole de commande IP – le module de protocole de commande de réseau du PPP – permettant de négocier les paramètres de protocole Internet souhaités. Pour plus de détails sur IPCP, voir RFC 1332.

```
IPCP_options_type::= array   IPCP_options_type_element

IPCP_options_type_element::=     structure

{
IPCP_Option_Type:         unsigned,
IPCP_Option_Length:       unsigned,
IPCP_Option_Data:         CHOICE
{
        array                           [1] -- pour Pref-Peer-IP,
        -- chaque adresse IP est de type double-long-unsigned
        boolean                         [3] -- pour GAO et USIP,
        double-long-unsigned            [6] -- pour Pref-Local-IP,
        long-unsigned                   [18] -- pour IP-Comp-Prot
}
}
```

NOTE 1 Dans tous les cas, comme cela est spécifié dans la RFC 1332, le champ IPCP_Option_Length inclut la longueur totale des trois champs: IPCP_Option_Type, IPCP_Option_Length et IPCP_Option_Data.

Les IPCP_Option_Types pris en charge sont les suivants:

a) IP-Compression-Protocol (IP-Comp-Prot) IPCP_Option_Type = 2. Voir RFC 1332.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Cette option de configuration fournit une méthode pour négocier l'utilisation d'un protocole de compression spécifique. Par défaut, la compression n'est pas activée. Les valeurs possibles sont:

0x0000 – Aucune compression IP n'est utilisée (valeur par défaut);
0x002d – TCP/IP avec compression Van Jacobson (RFC 1332);
0x0003 – Compression d'en-tête robuste (ROHC) (RFC 3241);
0x0061 – Compression d'en-tête IP (RFC 2507, RFC 3544).

b) Preferred-Local-IP-Address (Pref-Local-I), IPCP_Option_Type = 3. Voir RFC 1332.

Cette option de configuration fournit une façon de négocier l'adresse IP à utiliser sur l'extrémité locale de la liaison. Elle permet à l'expéditeur de la Configure-Request d'énoncer quelle adresse IP est souhaitée ou de demander que l'homologue fournisse les informations. L'homologue peut fournir cette information en acquittant négativement (NAK) l'option et en retournant une adresse IP valide;

NOTE 2 Dans la RFC 1332, le nom du Type 3 de l'option est "IP-Address".

NOTE 3 Les types d'options 20, 21 et 22 ont été spécifiés pour les besoins de DLMS®/COSEM. Ils ne sont pas spécifiés dans la RFC 1332.

c) Preferred-Peer-IP-Addresses (Pref-Peer-IP), IPCP_Option_Type = 20.

Cette option de configuration fournit une façon de négocier l'adresse IP à utiliser sur l'extrémité distante de la liaison. Si la valeur TRUE est attribuée au paramètre Grant-Access-Only-to-Pref-Peer-on-List (GAO), le dispositif doit accepter la connexion PPP uniquement avec un dispositif distant dont l'une des adresses IP figure dans cette liste. Si la valeur TRUE est attribuée au paramètre Use-Static-IP-Pool (USIP), le dispositif COSEM serveur doit tenter d'attribuer l'une de ces adresses IP au dispositif distant.

d) Grant-Access-Only-to-Pref-Peer-on-List (GAO), IPCP_Option_Type = 21.

Cette option de configuration indique si le dispositif peut accepter la connexion PPP uniquement avec des dispositifs homologues dont l'adresse IP figure sur la liste ou non. Sa valeur par défaut est FALSE.

e) Use-Static-IP-Pool (USIP), IPCP_Option_Type = 22.

Cette option de configuration indique s'il convient ou non que le dispositif tente d'attribuer l'une des adresses IP des Preferred-Peer-IP-Addresses au dispositif situé à l'extrémité distante au cours de la phase de négociation d'adresses IP.

4.9.5.2.5 PPP_authentication

Contient les paramètres exigés par la procédure d'authentification de PPP utilisée.

```
PPP_auth_type::=      CHOICE

{
null-data     [0] -- utilisé si aucune authentification n'est exigée,
structure     [2] -- PAP_login or CHAP_algorithm or EAP_params
}

PAP_login::=      structure
{
user-name:        octet-string,
PAP-password:     octet-string
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
CHAP_algorithm ::=     structure
{
user-name:         octet-string,
algorithm_id:      unsigned
}
```

Les valeurs possibles pour le paramètre CHAP-algorithm-id sont aujourd'hui les suivantes:
0x05 – CHAP avec MD5 (par défaut) (voir RFC 1994),
0x06 – SHA-1,
0x80 – MS-CHAP (voir RFC 2433),
0x81 – MS-CHAP-2 (voir RFC 2759).

NOTE 1 De nouvelles valeurs peuvent être utilisées à mesure qu'elles sont attribuées.

Lorsque CHAP est utilisé, un "secret" est également exigé pour vérifier le "défi" envoyé par le client. Ce "secret" n'est pas accessible dans l'objet "PPP setup".

NOTE 2 Le protocole CHAP (protocole d'authentification par défi-réponse) PPP est spécifié dans la RFC 1994.

```
EAP_params::= structure

{
md5_challenge (Type 4):                    boolean,
one_time_password (OTP, Type 5):           boolean,
generic_token_card (GTC, Type 6):          boolean
}
```

NOTE 3 Le protocole EAP (protocole d'authentification extensible) est spécifié dans la RFC 3748.

4.9.6 SMTP setup (class_id = 46, version = 0)

4.9.6.1 Vue d'ensemble

Cette IC permet d'établir un échange de données en utilisant le protocole SMTP.

SMTP modem setup		**0...n**	**class_id = 46, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. server_port	(static)	long-unsigned			25	x + 0x08
3. user_name	(static)	octet-string				x + 0x10
4. login_password	(static)	octet-string				x + 0x18
5. server_address	(static)	octet-string				x + 0x20
6. sender_address	(static)	octet-string				x + 0x28
Méthodes spécifiques		***o/f***				

4.9.6.2 Description d'attribut

4.9.6.2.1 logical_name

Identifie l'instance de l'objet "SMTP setup". Voir 6.2.23.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.6.2.2 server_port

Définit la valeur du port TCP-UDP lié à ce protocole. Par défaut, cette valeur est le numberID de port SMTP attribué par l'IANA:

smtp 25/tcp, smtp 25/udp Simple Mail Transfer (transfert simple de courrier)

4.9.6.2.3 user_name

Définit le nom d'utilisateur à utiliser pour la connexion au serveur SMTP.

4.9.6.2.4 login_password

Mot de passe à utiliser pour la connexion. Si la chaîne est vide, cela signifie qu'il n'y a pas d'authentification.

4.9.6.2.5 server_address

Définit l'adresse de serveur sous la forme d'une chaîne d'octets. Cette adresse de serveur peut être un nom, qui doit pouvoir être résolu par le DNS primaire ou le DNS secondaire. S'il s'agit directement de l'adresse IP du serveur, qui est spécifiée ici, elle doit être une chaîne au format comportant des points. EXEMPLE: 163.187.45.87.

4.9.6.2.6 sender_address

Définit l'adresse de l'expéditeur sous la forme d'une chaîne d'octets. Cette adresse d'expéditeur peut être un nom. S'il s'agit directement de l'adresse IP de l'expéditeur, qui est spécifiée ici, il s'agira d'une chaîne au format comportant des points.

4.9.7 NTP setup (class_id = 100, version = 0)

4.9.7.1 Vue d'ensemble

Les instances de l'IC "NTP setup" permettent de configurer la synchronisation temporelle à l'aide du protocole NTP tel que spécifié dans la RFC 5905. Une ou plusieurs instances peuvent être configurées pour prendre en charge plusieurs serveurs de synchronisation.

NTP Setup		**0...n**	**class_id = 100, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. activated	(static)	boolean			FALSE	x + 0x08
3. server_address	(static)	octet-string				x + 0x10
4. server_port	(static)	long-unsigned			123	x + 0x18
5. authentication_method	(static)	enum				x + 0x20
6. authentication_keys	(static)	array				x + 0x28
7. client_key	(static)	octet-string				x + 0x30
Méthodes spécifiques		***o/f***				
1. synchronize (data)		o				x + 0x38
2. add_authentication_key (data)		f				x + 0x40
3. delete_authentication_key (data)		f				x + 0x48

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.7.2 Description d'attribut

4.9.7.2.1 Généralités

logical_name Identifie l'instance de l'objet "NTP setup". Voir 6.2.23.

4.9.7.2.2 activated

Définit si la synchronisation temporelle NTP est active ou non.

Synchronisation active = TRUE

4.9.7.2.3 server_address

Définit l'adresse de serveur NTP sous la forme d'une chaîne d'octets. Cette adresse de serveur peut être un nom, qui doit pouvoir être résolu par le DNS primaire ou le DNS secondaire. S'il s'agit directement de l'adresse IP du serveur, la notation décimale doit être utilisée pour les adresses IPv4, le format texte devant l'être pour les adresses IPv6.

Exemples: 163.187.45.87 (IPv4) ou 2001:db8::1:0:0:1 (IPv6)

4.9.7.2.4 server_port

Définit la valeur du port UDP associé à ce protocole. Par défaut, cette valeur est l'ID de numéro de port NTP attribué par l'IANA:

ntp 123/udp Network Time Protocol

4.9.7.2.5 authentication_ method

Définit le mode d'authentification utilisé pour le protocole NTP.

enum:

(0) no_security;
(1) shared_secrets;
(2) auto_key_IFF

4.9.7.2.6 authentication_keys

Contient les clés symétriques nécessaires si le mode de secrets partagés de l'authentification est utilisé.

authentication_keys::= array authentication_key

authentication_key::= structure

{
key_id double-long-unsigned,
key octet-string
}

4.9.7.2.7 client_key

Spécifie la clé client (clé publique du serveur NTP) pour le mécanisme d'authentification automatique de clé NTP utilisant le schéma d'authentification IFF (auto_key_IFF).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.9.7.3 Description de la méthode

4.9.7.3.1 synchronize (data)

Synchronise le temps du serveur DLMS® avec le serveur NTP.

data::= integer(0)

4.9.7.3.2 add_authentication_key (data)

Ajoute une nouvelle clé d'authentification symétrique au tableau de clé d'authentification.

data::= structure

{
key_id double-long-unsigned,
key octet-string
}

4.9.7.3.3 delete_authentication_key (data)

Supprime une clé d'authentification symétrique du tableau de clé. La clé à supprimer est identifiée par son key_id.

data::= double-long-unsigned

4.10 Classes d'interfaces pour l'établissement d'échange de données en utilisant le PLC à modulation S-FSK

4.10.1 Généralités

Ce paragraphe spécifie les classes d'interfaces COSEM pour établir et gérer les couches de protocoles du profil de communications DLMS®/COSEM S-FSK PLC:

- la couche physique S-FSK et la sous-couche MAC telles que définies dans l'IEC 61334-5-1:2001 et l'IEC 61334-4-512:2001;
- la sous-couche LLC telle que spécifiée dans l'IEC 61334-4-32:1996.

Les variables MIB/paramètres de liaison logique respectivement spécifiés dans l'IEC 61334-4-512:2001, l'IEC 61334-5-1:2001, l'IEC 61334-4-32:1996 et l'ISO/IEC 8802-2:1998 ont été mis en correspondance avec les attributs et/ou méthodes des IC de la spécification COSEM. La spécification de ces éléments a été reprise des normes ci-dessus et le texte a été adapté à l'environnement DLMS®/COSEM.

NOTE L'IEC 61334-4-512:2001 spécifie également un certain nombre de variables de gestion à utiliser du côté client. Cependant, le modèle d'objet du côté client n'est pas couvert par le présent document.

Pour les définitions relatives au profil S-FSK PLC, voir 3.2.

4.10.2 Vue d'ensemble

Les objets COSEM pour établir le canal PLC S-FSK et la couche LLC, si elle est mise en œuvre, doivent être situés dans le dispositif logique de gestion des serveurs COSEM.

La Figure 27 donne un exemple avec un dispositif physique COSEM comprenant trois dispositifs logiques. Chaque dispositif logique doit contenir un objet LDN (Nom de dispositif logique). Chaque dispositif logique contient un ou plusieurs objets d'association, un par client pris en charge.

NOTE Comme il y a plus d'un dispositif logique dans cet exemple, le dispositif logique de gestion obligatoire contient un objet "SAP Assignment" en lieu et place d'un objet LDN.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

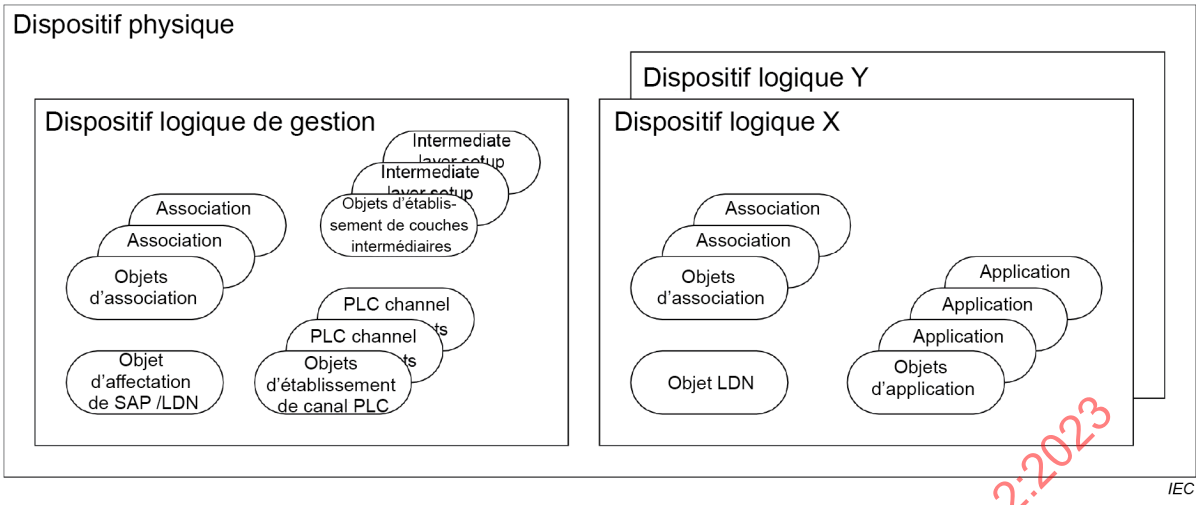


Figure 27 – Modèle d'objet des serveurs DLMS®/COSEM

Le dispositif logique de gestion contient les objets d'établissement des couches physiques et MAC du canal PLC, ainsi que des objets d'établissement pour la ou les couches intermédiaires. Il peut contenir des objets d'applications supplémentaires.

Les autres dispositifs logiques contiennent, en plus des objets d'association et LDN susmentionnés, d'autres objets d'applications, contenant des paramètres et des valeurs de mesure.

L'IEC 61334-4-512:2001 utilise des variables nommées selon la DLMS® pour modéliser les objets MIB et spécifie leur nom DLMS® dans la plage 8…184. Pour la compatibilité avec des mises en œuvre existantes, les noms courts 8…400 [sic] sont réservés aux dispositifs utilisant les profils PLC S-FSK de l'IEC 61334-5-1:2001 sans COSEM. Par conséquent, cette plage ne doit pas être utilisée lors de la mise en correspondance des attributs et méthodes des objets COSEM spécifiés dans le présent document, avec les variables nommées selon la DLMS® (mise en correspondance par SN).

Le Tableau 38 présente la mise en correspondance des variables MIB avec les attributs et/ou les méthodes des IC de la COSEM.

A noter d'une part que toutes les variables MIB spécifiées dans l'IEC 61334-4-512:2001 n'ont pas été mises en correspondance avec les attributs et méthodes des IC de COSEM. D'autre part, le présent document spécifie un certain nombre de nouvelles variables de gestion.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 38 – Mise en correspondance des variables MIB de l'IEC 61334-4-512:2001 avec les attributs/méthodes des IC COSEM

Nom	Référence (sauf indication contraire)	Classe d'interfaces	class_id/attribut/ méthode
Gestion de couche physique S-FSK			
delta-electrical-phase	variable 1	S-FSK Phy&MAC set-up (class_id = 50, version = 1)	50 / Attr. 3
max-receiving-gain	variable 2		50 / Attr. 4
max-transmitting-gain	–		50 / Attr. 5
search-initiator-threshold	–		50 / Attr. 6
frequencies	–		50 / Attr. 7
transmission-speed	–		50 / Attr. 15
Gestion de couche MAC			
mac-address	variable 3	S-FSK Phy&MAC set-up (class_id = 50, version = 1)	50 / Attr. 8
mac-group-addresses	variable 4		50 / Attr. 9
repeater	variable 5		50 / Attr. 10
repeater-status	–		50 / Attr. 11
search-initiator time-out	–	S-FSK MAC synchronization timeouts (class_id = 52, version = 0)	52 / Attr. 2
synchronization-confirmation-time-out	variable 6		52 / Attr. 3
time-out-not-addressed	variable 7		52 / Attr. 4
time-out-frame-not-OK	variable 8		52 / Attr. 5
min-delta-credit	variable 9	S-FSK Phy&MAC set-up (class_id = 50, version = 1)	50 / Attr. 12
initiator-mac-address	IEC 61334-5-1:2001 4.3.7.6		50 / Attr. 13
synchronization-locked	variable 10		50 / Attr. 14
Gestion de couche LLC IEC 61334-4-32			
max-frame-length	IEC 61334-4-32:1996 5.1.4	IEC 61334-4-32 LLC setup (class_id = 55, version = 1)	55 / Attr. 2
reply-status-list	variable 11		55 / Attr. 3
broadcast-list	variable 12	–	–
L-SAP-list	variable 13	NOTE Dans DLMS®/COSEM, les L-SAP des dispositifs logiques sont détenus par un objet "SAP Assignment".	
Gestion ACSE			
application-context-list	variable 14	NOTE Dans DLMS®/COSEM, les objets d'association jouent un rôle similaire.	
Gestion d'application			
active-initiator	variable 15	S-FSK Active initiator (class_id = 51, version = 0)	51 / Attr. 2
Objets système MIB			
reporting-system-list	variable 16	S-FSK Reporting system list (class_id = 56, version = 0)	56 / Attr. 2

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Nom	Référence (sauf indication contraire)	Classe d'interfaces	class_id/attribut/ méthode
Autres objets MIB			
reset-NEW-not-synchronized	variable 17	S-FSK Active initiator (class_id = 51, version = 0)	51 / Méthode 1
new-synchronization	IEC 61334-5-1:2001 4.3.7.6	–	
initiator-electrical-phase	variable 18		50 / Attr. 2
broadcast-frames-counter	variable 19	S-FSK MAC counters (class_id = 53, version = 0)	53 / Attr. 4
repetitions-counter	variable 20		53 / Attr. 5
transmissions-counter	variable 21		53 / Attr. 6
CRC-OK-frames-counter	variable 22		53 / Attr. 7
CRC-NOK-frames-counter	–		53 / Attr. 8
synchronization-register	variable 23		53 / Attr. 2
desynchronization-listing	variable 24		53 / Attr. 3

4.10.3 S-FSK Phy&MAC set-up (class_id = 50, version = 1)

4.10.3.1 Vue d'ensemble

NOTE L'utilisation de la version 0 de cette classe d'interfaces est déconseillée.

Une instance de la classe "S-FSK Phy&MAC set-up" stocke les données nécessaires pour établir et gérer la couche physique et la couche MAC du profil de couche inférieure PLC S-FSK.

S-FSK Phy&MAC setup		**0...n**	**class_id = 50, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. initiator_electrical_phase	(static)	enum	0	3		x + 0x08
3. delta_electrical_phase	(dyn.)	enum	0	6		x + 0x10
4. max_receiving_gain	(static)	unsigned				x + 0x18
5. max_transmitting_gain	(static)	unsigned				x + 0x20
6. search_initiator_threshold	(static)	unsigned			98	x + 0x28
7. frequencies	(static)	frequencies_type				x + 0x30
8. mac_address	(dyn.)	long-unsigned			FFE	x + 0x38
9. mac_group_addresses	(static)	array				x + 0x40
10. repeater	(static)	enum				x + 0x48
11. repeater_status	(dyn.)	boolean				x + 0x50
12. min_delta_credit	(dyn.)	unsigned				x + 0x58
13. initiator_mac_address	(dyn.)	long-unsigned				x + 0x60
14. synchronization_locked	(dyn.)	boolean				x + 0x68
15. transmission_speed	(static)	enum	0	6	3	x + 0x70
Méthodes spécifiques		***o/f***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2 Description d'attribut

4.10.3.2.1 logical_name

Identifie l'instance de l'objet "S-FSK Phy&MAC setup". Voir 6.2.24.

4.10.3.2.2 initiator_electrical_phase

Contient la variable MIB initiator-electrical-phase (variable 18) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

Il est inscrit par le système client pour indiquer la phase à laquelle il est connecté.

enum:

(0) non définie (par défaut);
(1) phase 1;
(2) phase 2;
(3) phase 3.

NOTE Cette énumération diffère de celle de l'IEC 61334-4-512.

4.10.3.2.3 delta_electrical_phase

Contient la variable MIB *delta-electrical-phase* (variable 1) spécifiée dans l'IEC 61334-4-512:2001, 5.2, et l'IEC 61334-5-1:2001, 3.5.5.3.

Il indique la différence de phase entre la phase de connexion du client et la phase de connexion du serveur. Les valeurs suivantes sont prédéfinies:

enum:

(0) Non définie: le serveur est temporairement incapable de déterminer la différence de phase,
(1) Le système serveur est connecté à la même phase que le système client.

La différence de phase entre la phase de connexion du serveur et la phase de connexion du client est égale à:

(2) 60 degrés;
(3) 120 degrés;
(4) 180 degrés;
(5) -120 degrés;
(6) -60 degrés.

4.10.3.2.4 max_receiving_gain

Contient la variable MIB *max-receiving-gain* (variable 2) spécifiée dans l'IEC 61334-4-512:2001, 5.2, et l'IEC 61334-5-1:2001, 3.5.5.3.

Correspond au gain autorisé maximal destiné à être utilisé par le système serveur dans le mode réception. L'unité par défaut est dB.

NOTE Dans l'IEC 61334-4-512:2001, aucune unité n'est spécifiée.

Les valeurs possibles du gain peuvent dépendre du matériel. Par conséquent, après écriture d'une valeur dans cet attribut, il convient de relire la valeur pour connaître la valeur exacte.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2.5 max_transmitting_gain

Contient la valeur du *max-transmitting-gain*.

Correspond à l'affaiblissement maximal destiné à être utilisé par le système serveur dans le mode émission. L'unité par défaut est dB.

Les valeurs possibles du gain peuvent dépendre du matériel. Par conséquent, après écriture d'une valeur dans cet attribut, il convient de relire la valeur pour connaître la valeur exacte.

4.10.3.2.6 search_initiator_threshold

Cet attribut est utilisé dans le processus d'initiateur de recherche intelligente. Si la valeur du signal de l'initiateur se situe au-dessus de la valeur de cet attribut, un processus de synchronisation rapide est possible.

La valeur par défaut est 98 dBµV.

4.10.3.2.7 frequencies

Contient les fréquences exigées par la modulation S-FSK.

```
frequencies_type::= structure

{
mark_frequency:        double-long-unsigned,
space_frequency:       double-long-unsigned
}
```

L'unité par défaut est Hz.

4.10.3.2.8 mac_address

Contient la variable MIB *mac-address* (variable 3) spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

NOTE 1 Les adresses MAC sont exprimées sur 12 bits.

Contient la valeur de l'adresse de la jonction physique (adresse MAC) associée au système local. Dans l'état non configuré, l'adresse MAC est "NEW-address".

L'attribut est écrit en local par le CIASE lorsque le système est enregistré (auprès d'un service Register). La valeur est utilisée dans chaque trame sortante ou entrante. La valeur par défaut est "NEW-address".

Cet attribut est défini sur NEW:

– par la sous-couche MAC, une fois que le retard time-out-not-addressed est dépassé;
– lorsqu'un système client "réinitialise" le système serveur. Voir 4.10.4.

Lorsque cet attribut est défini sur NEW:

– le système perd sa synchronisation (fonction de la sous-couche MAC);
– l'attribut *mac_group_address* est réinitialisé (tableau de 0 élément);
– le système libère automatiquement toutes les AA qui peuvent l'être.

NOTE 2 Le deuxième point n'est pas présent dans l'IEC 61334-4-512:2001.

Les adresses MAC prédéfinies sont indiquées dans le Tableau 39.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2.9 mac_group_addresses

Contient la variable MIB *mac-group-address* (variable 4) spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

Contient un jeu d'adresses de groupe MAC utilisées à des fins de diffusion.

array mac-address
mac-address::= long-unsigned

Les adresses ALL-configured-address, ALL-physical-address et NO-BODY ne sont pas incluses dans cette liste. Celles-ci sont des valeurs prédéfinies internes.

Cet attribut doit être écrit par l'initiateur utilisant les services DLMS® pour déclarer des adresses de groupe MAC spécifiques sur un système serveur.

Cet attribut est lu en local par la sous-couche MAC lors de la vérification du champ d'adresse de destination d'une trame MAC non reconnue comme adresse individuelle ou comme l'une des trois valeurs prédéfinies (ALL-configured-address, ALL-physical-address et NO-BODY).

4.10.3.2.10 repeater

Contient la variable MIB *repeater* (variable 5) spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

Il spécifie si le système serveur répète effectivement toutes les trames ou non.

enum:

(0) never repeater;
(1) always repeater;
(2) dynamic repeater.

Si la variable *repeater* est égale à 0, il convient que le système serveur ne répète jamais les trames.

Si sa valeur est 1, le système serveur est un répéteur: il doit répéter toutes les trames reçues sans erreur et avec un crédit courant supérieur à zéro.

Si sa valeur est 2, l'état de répéteur peut être modifié de façon dynamique par le serveur lui-même.

NOTE La valeur 2 n'est pas spécifiée dans l'IEC 61334-4-512.

Cet attribut est lu en interne par la sous-couche MAC chaque fois qu'une trame est reçue.

La valeur par défaut doit être spécifiée dans des spécifications d'accompagnement spécifiques au projet.

4.10.3.2.11 repeater_status

Contient l'état de répéteur (*repeater status*) en cours du dispositif.

boolean:

(0) FALSE = pas de répéteur;
(1) TRUE = répéteur.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.3.2.12 min_delta_credit

Contient la variable MIB *min-delta-credit* (variable 9) spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

NOTE Seuls les trois bits de poids faible sont utilisés.

Le Delta Credit (DC) est la soustraction des champs Initial Credit (IC) et Current Credit (CC) d'une trame MAC reçue correcte. La valeur minimale de delta-credit pour une trame MAC reçue correcte, dirigée vers un système serveur, est contenue dans cette variable.

La valeur par défaut est mise au crédit initial maximal (voir IEC 61334-5-1:2001, 4.2.3.1 pour de plus amples explications concernant le crédit et la valeur de MAX_INITIAL_CREDIT). Un système client peut réinitialiser cette variable en mettant cette valeur au crédit initial maximal.

4.10.3.2.13 initiator_mac_address

Contient la variable MIB *initiator-mac-address* spécifiée dans l'IEC 61334-5-1:2001, 4.3.7.6.

Sa valeur est soit l'adresse MAC de l'active-initiator, soit l'adresse NO-BODY, en fonction de la valeur de l'attribut *synchronization_locked* (voir 4.10.3.2.14). Voir aussi IEC 61334-5-1:2001, 3.5.3, 4.1.6.3 et 4.1.7.2.

4.10.3.2.14 synchronization_locked

Contient la variable MIB *synchronization-locked* (variable 10) spécifiée dans l'IEC 61334-4-512:2001, 5.3.

Commande l'état verrouillé/déverrouillé (locked/unlocked) de la synchronisation. Voir IEC 61334-5-1:2001 pour plus de détails.

Si la valeur de cet attribut est égale à TRUE, le système est à l'état synchronization-locked. Dans cet état, l'*initiator-mac-address* est toujours égale au champ d'adresse MAC de l'objet MIB active-initiator. l'attribut 2 de IC "S-FSK Active initiator" (4.10.4.2.2).

Si la valeur de cet attribut est FALSE, le système est à l'état synchronization-unlocked. Dans cet état, l'attribut *initiator_mac_address* est toujours mis à la valeur NO-BODY: un changement de valeur dans le champ d'adresse MAC de l'objet MIB active-initiator n'a pas d'incidence sur le contenu de l'attribut *initiator_mac_address*, qui reste à la valeur NO-BODY. La valeur par défaut de cette variable doit être spécifiée dans les spécifications de mise en œuvre.

NOTE À l'état synchronization-unlocked, le serveur se synchronise sur n'importe quelle trame valide. À l'état synchronization-locked, le serveur se synchronise seulement sur les trames produites ou dirigées vers le système client dont l'adresse MAC est égale à la valeur de l'attribut initiator_mac_address.

4.10.3.2.15 transmission_speed

La vitesse de transmission prise en charge par le dispositif physique. Voir également IEC 61334-5-1:2001, 3.2.2.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

enum:	50 Hz	60 Hz
(0)	300 bauds	360 bauds
(1)	600 bauds	720 bauds
(2)	1 200 bauds	1 440 bauds
(3) -- par défaut	2 400 bauds	2 880 bauds
(4)	4 800 bauds	5 760 bauds
(5)	7 200 bauds	8 640 bauds
(6)	9 600 bauds	11 520 bauds

Tableau 39 – Adresses MAC dans le profil S-FSK

Adresse	Valeur
NO-BODY	000
Local MAC	001...FIMA-1
Initiator	FIMA...LIMA
MAC group address	LIMA + 1...FFB
All configured	FFC
NEW	FFE
All Physical	FFF
NOTE Les adresses MAC sont exprimées sur 12 bits. Ces adresses sont spécifiées dans l'IEC 61334-5-1:2001, 4.2.3.2, 4.3.7.5.1, 4.3.7.5.2 et 4.3.7.5.3. FIMA: adresse MAC du premier initiateur; C00. LIMA: adresse MAC du dernier initiateur; DFF.	

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.4 S-FSK Active initiator (class_id = 51, version = 0)

4.10.4.1 Vue d'ensemble

Une instance de l'IC "S-FSK Active initiator" stocke les données de l'initiateur actif. L'initiateur actif est le système client, qui a enregistré en dernier le système serveur avec une demande d'enregistrement CIASE. Voir IEC 61334-4-511:2000, 7.2.

S-FSK Active initiator		0...n	class_id = 51, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. active_initiator	(dyn.)	initiator_descriptor				x + 0x08
Méthodes spécifiques		***o/f***				
1. reset_NEW_not_synchronized (data)						x + 0x10

4.10.4.2 Description d'attribut

4.10.4.2.1 logical_name

Identifie l'instance de l'objet "S-FSK Active initiator". Voir 6.2.25.

4.10.4.2.2 active_initiator

Contient la variable MIB *active-initiator* (variable 15) spécifiée dans l'IEC 61334-4-512:2001, 5.6.

Contient les identifiants du dernier initiateur actif qui a enregistré le système avec une demande d'enregistrement. Voir IEC 61334-4-511:2000, 7.2.

Le système initiateur est identifié par son titre de système, son adresse MAC et son sélecteur de L-SAP:

initiator_descriptor::= structure

{
system_title: octet-string,
MAC_address: long-unsigned,
L_SAP_selector: unsigned
}

La taille et la structure du titre de système peuvent être spécifiées dans les spécifications du système. Lorsque le titre de système est utilisé comme partie intégrante du vecteur d'initialisation d'algorithmes cryptographiques, la taille doit satisfaire aux exigences applicables au vecteur d'initialisation.

L'élément MAC_address est utilisé pour mettre à jour la variable de gestion MAC *initiator-mac-address* lorsque le système est configuré à l'état synchronization-locked. Voir la spécification des attributs *initiator_mac_address* et *synchronization_locked* de l'IC "S-FSK Phy&MAC setup" en 4.10.3.

Tant que le serveur n'est pas enregistré par un initiateur actif, le champ L_SAP_selector est mis à 0 et le champ system_title est égal à une chaîne d'octet de zéros (0).

La valeur par défaut de l'initiator-descriptor est: system_title = octet-string de zéros, MAC_address = NO-BODY et L_SAP_selector = 0.

La valeur de cet attribut peut être mise à jour par l'appel de la méthode *reset_NEW_not_synchronized* ou par le service "CIASE Register".

4.10.4.3 Description de la méthode

4.10.4.3.1 reset_NEW_not_synchronized (data)

Contient la variable MIB *reset-NEW-not-synchronized* (variable 17) spécifiée dans l'IEC 61334- 4-512:2001, 5.8.

Permet à un système client de "réinitialiser" le système serveur. La valeur proposée correspond à une adresse MAC de client. L'écriture est refusée si:

– la valeur ne correspond pas à une adresse MAC de client valide ou à l'adresse NO-BODY prédéfinie;
– la valeur proposée est différente de l'adresse NO-BODY et l'attribut *synchronization_locked* n'est pas égal à TRUE.

Pour la description du processus d'initiateur de recherche intelligente, voir IEC 62056-8-3:2013, 10.7.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Lorsque cette méthode est appelée, les actions suivantes sont exécutées:

– le système retourne à l'état non configuré (UNC: MAC-address est égale à NEW-address). Cette transition provoque automatiquement la perte de synchronisation (fonction de la sous-couche MAC);
– le système modifie la valeur de l'attribut *active_initiator*: la MAC_address est mise à la valeur proposée, le L-SAP_selector est mis à la valeur 0 et le system_title est mis à une chaîne d'octets de zéros;
– toutes les AA qui peuvent être libérées le sont.

4.10.5 S-FSK MAC synchronization timeouts (class_id = 52, version = 0)

4.10.5.1 Vue d'ensemble

Une instance de l'IC "S-FSK MAC synchronization timeouts" stocke les temps d'expirations liées au processus de synchronisation.

S-FSK MAC synchronization timeouts		**0...n**	**class_id = 52, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. search_initiator_timeout	(static)	long-unsigned				x + 0x08
3. synchronization_confirmation_ timeout	(static)	long-unsigned				x + 0x10
4. time_out_not_addressed	(static)	long-unsigned				x + 0x18
5. time_out_frame_not_OK	(static)	long-unsigned				x + 0x20
Méthodes spécifiques		***o/f***				

4.10.5.2 Description d'attribut

4.10.5.2.1 logical_name

Identifie l'instance de l'objet "S-FSK synchronization timeouts". Voir 6.2.25.

4.10.5.2.2 search_initiator_timeout

Ce temps d'expiration prend en charge la fonction d'initiateur de recherche intelligente.

Il définit la valeur du temps, en secondes, pendant lequel le système serveur recherche l'initiateur ayant le signal le plus fort.

Au cours du temps d'expiration, tous les initiateurs, qui peuvent être entendus par les serveurs, sont censés parler.

Après expiration de ce temps, le serveur acceptera une demande d'enregistrement issue de l'initiateur ayant le signal le plus fort et verrouillera sur l'initiateur en question.

Si la valeur du temps d'expiration est égale à 0, cela signifie que la caractéristique n'est pas utilisée.

Le temps d'expiration est lancé au début de la phase de recherche d'initiateur, lorsque le serveur reçoit la première trame avec une adresse MAC d'initiateur valide. Le temps d'expiration est relancé lorsque la phase de recherche d'initiateur est terminée et que le serveur verrouille sur l'initiateur. Au cours de la phase de vérification d'initiateur, elle est relancée à la réception de chaque trame valide.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Une synchronisation rapide peut être accomplie si le niveau de signal est suffisamment bon (Niveau du signal d'initiateur >= Search-Initiator-Threshold) et si l'une des adresses MAC (Source ou Destination) est une adresse MAC d'initiateur. Cela signifie que le module (le compteur) est proche d'un DC ou proche d'un module qui est déjà verrouillé sur le DC en question. Le module verrouille dans ce cas sur l'initiateur en question.

4.10.5.2.3 synchronization_confirmation_timeout

Contient la variable MIB *synchronization-confirmation-timeout* (variable 6) spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

Définit la valeur du temps, en secondes, après lequel un système serveur qui devient juste synchronisé à une trame (détection d'un chemin de données égal à AAAA54C7 hex) perdra automatiquement sa synchronisation de trame si la sous-couche MAC n'identifie pas une trame MAC valide. Le temps d'expiration démarre après la réception des quatre premiers octets d'une trame physique.

La valeur de cette variable peut être modifiée par un système client. Ce temps d'expiration assure une rapide désynchronisation d'un système qui s'est synchronisé sur une trame physique incorrecte. Voir IEC 61334-5-1:2001, 3.5.3 pour plus de détails.

Il convient de spécifier la valeur par défaut de cette variable dans les spécifications de mise en œuvre.

Une valeur égale à 0 équivaut à annuler l'utilisation du compteur de *synchronization_confirmation_timeout* associé.

4.10.5.2.4 time_out_not_addressed

Contient la variable MIB *time-out-not-addressed* (variable 7) spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

Définit le temps, en minutes, après lequel un système serveur qui n'a pas été adressé individuellement:

– retourne à l'état non configuré (UNC: MAC-address est égale à NEW-address): cette transition implique automatiquement la perte de synchronisation (fonction de la sous-couche MAC) et la libération de toutes les AA qui peuvent l'être;
– perd son initiateur actif: la MAC_address de l'initiateur actif est mise à NO-BODY, le sélecteur LSAP est mis à la valeur 00 et le titre de système est mis à une chaîne d'octets de zéros.

Les adresses de diffusion n'étant pas des adresses système individuelles, le temporisateur associé au retard *time-out-not-addressed* assure qu'un système oublié retournera tôt ou tard à l'état non configuré. Il sera alors redécouvert.

Un système oublié est un système qui n'a pas été adressé individuellement pendant un temps supérieur à la quantité de temps "*time-out-not-addressed*".

Il convient de spécifier la valeur par défaut de cette variable dans les spécifications de mise en œuvre.

Une valeur égale à 0 équivaut à annuler l'utilisation du compteur de time-out-not-addressed associé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.5.2.5 time_out_frame_not_OK

Contient la variable MIB *time-out-frame-not-OK* (variable 8), spécifiée dans l'IEC 61334-4-512:2001, 5.3, et l'IEC 61334-5-1:2001, 4.3.7.6.

Définit le temps, en secondes, après lequel un système serveur qui n'a pas reçu une trame MAC correctement formée (champ NS incorrect, nombre incohérent de sous-trames reçues, faux contrôle de code de redondance cyclique) perd sa synchronisation de trame.

La valeur par défaut de cette variable doit être spécifiée dans les spécifications de mise en œuvre.

Une valeur égale à 0 équivaut à annuler l'utilisation du compteur de *time-out-frame-not-OK* associé.

4.10.6 S-FSK MAC counters (class_id = 53, version = 0)

4.10.6.1 Vue d'ensemble

Une instance de l'IC "S-FSK MAC counters" stocke les compteurs liés aux phases d'échange, d'émission et de répétition de trames.

S-FSK MAC counters		**0...n**	**class_id = 53, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. synchronization_register	(dyn.)	array				x + 0x08
3. desynchronization_listing	(dyn.)	structure				x + 0x10
4. broadcast_frames_counter	(dyn.)	array				x + 0x18
5. repetitions_counter	(dyn.)	double-long-unsigned			0	x + 0x20
6. transmissions_counter	(dyn.)	double-long-unsigned			0	x + 0x28
7. CRC_OK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x30
8. CRC_NOK_frames_counter	(dyn.)	double-long-unsigned				x + 0x38
Méthodes spécifiques		***o/f***				
1. reset (data)						x + 0x50

4.10.6.2 Description d'attribut

4.10.6.2.1 logical_name

Identifie l'instance de l'objet "S-FSK MAC counters". Voir 6.2.25.

4.10.6.2.2 synchronization_register

Contient la variable MIB *synchronization-register* (variable 23) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

```
array synchronization_couples

synchronization_couples::= structure

{
mac_address:                long-unsigned,
synchronizations_counter:   double-long-unsigned
}--
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Cette variable compte le nombre de processus de synchronisation exécutés par le système. Les processus qui conduisent à une perte de synchronisation en raison de la détection d'un initiateur erroné sont enregistrés. Les autres processus qui conduisent à une perte de synchronisation (temps d'expiration, écriture de gestion) ne sont pas enregistrés.

Cette variable fournit un bilan des différents systèmes sur lesquels le système serveur est "potentiellement" capable de se synchroniser.

Un processus de synchronisation est initialisé lorsque l'entité d'application de gestion (gestionnaire de connexion) reçoit une primitive MA_Sync.indication (Synchronization State = SYNCHRO_FOUND) provenant de l'entité de sous-couche MAC. Ce processus est enregistré dans la variable synchronization-register uniquement si la primitive MA_Sync.indication (Synchronization State = SYNCHRO_FOUND) est suivie de l'une des trois primitives:

1) primitive MA_Data.indication (DA, SA, MSDU);
2) MA_Sync.indication (Synchronization State = SYNCHRO_CONF, SA, DA);
3) MA_Sync.indication (Synchronization State = SYNCHRO_LOSS, Synchro Loss Cause = wrong_initiator, SA, DA.

NOTE La troisième primitive n'est générée que si le système serveur est configuré à l'état synchronization-locked. Voir 4.10.3.

Les processus qui conduisent à la génération de primitives MA_Sync.indication (Synchronization State = SYNCHRO_LOSS) indiquant une perte de synchronisation due:

– à la couche physique;
– au compteur time-out-not-addressed;
– à la mise de l'attribut mac_address de l'objet "S-FSK Phy&MAC setup" à NEW (voir 4.10.3);
– à l'appel de la méthode reset_NEW_not_synchronized de l'objet "S-FSK Active initiator" (également appelé écriture de gestion) (voir 4.10.4);

ne sont pas pris en compte dans cette variable.

Pour les détails relatifs à la primitive de service MA_Sync.indication, voir IEC 61334-5-1:2001, 4.1.7.1.

Si le processus de synchronisation se termine avec l'une des trois primitives énumérées ci-dessus, la variable "synchronization-register" est mise à jour en tenant compte des champs SA et DA de la primitive.

La variable *synchronization-register* est mise à jour comme suit:

En premier lieu, l'entité de gestion vérifie les champs SA et DA.

– Si l'un de ces champs correspond à une adresse MAC de client (CMA), l'entité:
  - vérifie si l'adresse MAC de client (CMA) apparaît dans l'un des couples contenus dans la variable synchronization-register;
  - si elle y apparaît, le sous-champ synchronizations-counter correspondant est incrémenté;
  - si elle n'y apparaît pas, un nouveau couple (mac-address, synchronizations-counter) est ajouté. Ce couple est initialisé à la valeur (CMA, 1).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

– Si aucun des champs SA et DA ne correspond à une adresse MAC de client, il est pris pour hypothèse que le système a trouvé sa référence de synchronisation sur une trame de type DiscoverReport. Dans ce cas, la mac-address qu'il convient d'enregistrer dans la variable synchronization-register est la valeur NEW prédéfinie (FFE). La mise à jour de la variable "synchronization-register" est accomplie de la même manière que pour une adresse MAC de client (CMA) normale.

Lorsqu'un champ *synchronizations-counter* atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément.

Il convient de spécifier dans les spécifications de mise en œuvre le nombre maximal de couples de synchronisation {mac-address, synchronizations-counter} contenus dans cette variable. Lorsque le maximum est atteint, la mise à jour de la variable suit un mécanisme de premier entré, premier sorti (FIFO): seules les adresses MAC sources les plus récentes sont mémorisées.

La valeur par défaut de cette variable est un tableau vide.

4.10.6.2.3 desynchronization_listing

Contient la variable MIB *desynchronization-listing* (variable 24) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

```
structure

{
nb_physical_layer_desynchronization:              double-long-unsigned,
nb_time_out_not_addressed_desynchronization:      double-long-unsigned,
nb_timeout_frame_not_OK desynchronization:        double-long-unsigned,
nb_write_request_desynchronization:               double-long-unsigned,
nb_wrong_initiator_desynchronization:             double-long-unsigned
}
```

Cette variable compte le nombre de désynchronisations qui se sont produites en fonction de leur cause. A la réception d'une notification de perte de synchronisation, l'entité de gestion met à jour cet attribut en incrémentant le compteur en fonction de la cause de la désynchronisation.

Lorsque l'un des compteurs atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément.

La valeur par défaut de cette variable est une structure dont tous les éléments sont égaux à 0.

4.10.6.2.4 broadcast_frames_counter

Contient la variable MIB *broadcast-frames-counter* (variable 19) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

```
array        broadcast-couples

broadcast-couples::= structure

{
source-mac-address:   long-unsigned,
frames-counter:       double-long-unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Il compte les trames de diffusion reçues par le système serveur et produites par un système client (source-mac-address = n'importe quelle client-mac-address valide, destination-mac address = ALL-physical). Le nombre de trames est classé en fonction de l'origine de l'émetteur. Le compteur est incrémenté même si la LLC-destination-address n'est pas valide sur le système serveur. Lorsqu'un champ frames-counter atteint sa valeur maximale, il retourne automatiquement à 0 sur le prochain incrément.

Il convient de spécifier dans les spécifications de mise en œuvre le nombre maximal de couples de diffusion "broadcast-couples" {source-mac-address, frames-counter} contenus dans cette variable. Lorsque le maximum est atteint, la mise à jour de la variable suit un mécanisme de premier entré, premier sorti (FIFO): seules les adresses MAC sources les plus récentes sont mémorisées.

4.10.6.2.5 repetitions_counter

Contient la variable MIB *repetitions-counter* (variable 20) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

Compte le nombre de phases de répétition. Les phases de répétition suivant une émission ne sont pas comptées. Si la sous-couche MAC est configurée dans le mode "non-répéteur" (no-repeater), cette variable n'est pas mise à jour. Le compteur de répétitions mesure l'activité du système comme un répéteur. Une trame reçue répétée cinq fois (de CC=4 à CC=0) est comptée seulement une seule fois dans le compteur de répétitions, car elle correspond à une phase de répétition. Le compteur est incrémenté au début de chaque phase de répétition. Lorsque le compteur de répétitions atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément. La valeur par défaut est 0.

4.10.6.2.6 transmissions_counter

Contient la variable MIB *transmissions-counter* (variable 21) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

Compte le nombre de phases d'émission. Une phase d'émission est caractérisée par l'émission et la répétition d'une trame. Une phase de répétition qui suit la réception d'une trame n'est pas comptée. Le compteur d'émissions est incrémenté au début de chaque phase d'émission. Un système client peut écrire cette variable pour mettre à jour le compteur. Lorsque le compteur d'émissions atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément. La valeur par défaut est 0.

4.10.6.2.7 CRC_OK_frames_counter

Contient la variable MIB *CRC-OK-frames-counter* (variable 22) spécifiée dans l'IEC 61334-4-512:2001, 5.8.

Compte le nombre de trames reçues avec un champ Frame Check Sequence correct. Lorsque le champ "CRC OK frames counter" atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément. La valeur par défaut est 0.

4.10.6.2.8 CRC_NOK_frames_counter

Compte le nombre de trames reçues avec un champ Frame Check Sequence incorrect. Lorsque le champ "CRC NOK frames counter" atteint la valeur maximale, il retourne automatiquement à 0 sur le prochain incrément. La valeur par défaut est 0.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.10.6.3 Description de la méthode

4.10.6.3.1 reset (data)

Efface tous les compteurs.

data::= integer (0)

4.10.7 IEC 61334-4-32 LLC setup (class_id = 55, version = 1)

4.10.7.1 Vue d'ensemble

Une instance de l'IC "IEC 61334-4-32 LLC setup" contient les paramètres nécessaires pour établir et gérer la couche LLC comme spécifié dans l'IEC 61334-4-32.

IEC 61334-4-32 LLC setup		**0...n**	**class_id = 55, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. max_frame_length	(static)	long-unsigned				x + 0x08
3. reply_status_list	(dyn.)	array				x + 0x10
Méthodes spécifiques		***o/f***				

4.10.7.2 Description d'attribut

4.10.7.2.1 logical_name

Identifie l'instance de l'objet "IEC 61334-4-32 LLC setup". Voir 6.2.25.

4.10.7.2.2 max_frame_length

Contient la longueur de la trame de LLC en octets. Voir IEC 61334-4-32:1996, 5.1.4.

Pour le profil PLC S-FSK, les valeurs minimales/par défaut/ maximales sont respectivement 26/134/242 octets. Voir IEC 61334-5-1:2001, 4.2.2.

NOTE Pour d'autres profils de couches inférieures, voir les valeurs correspondantes dans la spécification pertinente.

4.10.7.2.3 reply_status_list

Contient la variable MIB *reply-status-list* (variable 11) spécifiée dans l'IEC 61334-4-512:2001, 5.4.

Énumère les L-SAP qui ont un tampon RDR (Données de réponse sur demande) non vide, qui n'a pas été déjà lu. La longueur d'une L-SDU en attente est spécifiée en nombre de sous-trames (différent de zéro). La variable est générée en local par la sous-couche LLC.

```
array        reply_status

reply_status::= structure

{
L_SAP_selector:              unsigned,
length_of_waiting_L_SDU:     unsigned
}
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

length_of_waiting_L_SDU dans le cas du profil S-FSK est en nombre de sous-trames. Les valeurs valides sont 1 à 7.

4.10.8 S-FSK Reporting system list (class_id = 56, version = 0)

4.10.8.1 Vue d'ensemble

Une instance de l'IC "S-FSK Reporting system list" contient la liste des systèmes de comptes-rendus.

S-FSK Reporting system list		**0...n**	**class_id = 56, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. reporting_system_list	(dyn.)	array				x + 0x08
Méthodes spécifiques		***o/f***				

4.10.8.2 Description d'attribut

4.10.8.2.1 logical_name

Identifie l'instance de l'objet "S-FSK Reporting system list". Voir 6.2.25.

4.10.8.2.2 reporting_system_list

Contient la variable MIB *reporting-system-list* (variable 16) spécifiée dans l'IEC 61334-4-512:2001, 5.7.

array system-title
system-title::= octet-string

Contient les titres des systèmes serveurs qui ont fait une demande de DiscoverReport et qui n'ont pas été déjà enregistrés. La liste a une taille finie et est triée à l'arrivée. Le premier élément est le plus récent. Une fois qu'elle est pleine, les éléments les plus anciens sont remplacés par les nouveaux.

La liste de systèmes de comptes-rendus est mise à jour:

– lorsqu'une CI_PDU DiscoverReport est reçue par le système serveur (quel que soit son état: non configuré ou configuré): le CIASE ajoute le titre de système de notification au début de la liste et vérifie qu'il n'existe nulle part ailleurs dans la liste; s'il existe, il détruit l'ancien. Un titre de système ne peut exister qu'une seule fois dans la liste;
– lorsqu'une CI_PDU Register est reçue par le système serveur (quel que soit son état: non configuré ou configuré): le CIASE vérifie la liste de systèmes de comptes-rendus. Si un titre de système est présent dans la liste de systèmes de comptes-rendus et dans la CI-PDU Register, le CIASE supprime le titre de système de la liste de systèmes de comptes-rendus: ce système n'est plus considéré comme un système de notification.

4.11 Classes d'interfaces pour l'établissement de la couche LLC selon l'ISO/IEC 8802-2

4.11.1 Généralités

Le présent paragraphe spécifie les IC disponibles pour l'établissement de la couche LLC selon l'ISO/IEC 8802-2, utilisée dans certains profils de communication DLMS®/COSEM, dans les divers types de fonctionnement.

Pour les définitions liées à la couche LLC selon l'ISO/IEEE 8802-2, voir ISO/IEC 8802-2:1998, 1.4.2.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.11.2 ISO/IEC 8802-2 LLC Type 1 setup (class_id = 57, version = 0)

4.11.2.1 Vue d'ensemble

Une instance de l'IC "ISO/IEC 8802-2 LLC Type 1 setup" contient les paramètres nécessaires pour établir la couche LLC selon l'ISO/IEC 8802-2 dans le fonctionnement de type 1.

ISO/IEC 8802-2 LLC Type 1 setup		**0...n**	**class_id = 57, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. max_octets_ui_pdu	(static)	long unsigned			128	x + 0x08
Méthodes spécifiques		***o/f***				

4.11.2.2 Description d'attribut

4.11.2.2.1 logical_name

Identifie l'instance de l'objet "ISO/IEC 8802-2 LLC Type 1 setup". Voir 6.2.26.

4.11.2.2.2 max_octets_ui_pdu

Voir la spécification de protocole MAC appropriée pour toute limitation du nombre maximal d'octets dans une PDU UI. Aucune restriction n'est imposée par la sous-couche LLC. Cependant, dans l'intérêt de disposer d'une valeur sur laquelle tous les utilisateurs de LLC de type 1 peuvent s'appuyer, tous les MAC doivent au moins être capables de prendre en charge les PDU UI avec des champs d'informations d'une longueur maximale pouvant atteindre 128 octets.

Voir ISO/IEC 8802-2:1998, 6.8.1, *Nombre maximal d'octets* dans une PDU UI.

4.11.3 ISO/IEC 8802-2 LLC Type 2 setup (class_id = 58, version = 0)

4.11.3.1 Vue d'ensemble

Une instance de l'IC "ISO/IEC 8802-2 LLC Type 2 setup" contient les paramètres nécessaires pour établir la couche LLC selon l'ISO/IEC 8802-2 dans le fonctionnement de type 2.

ISO/IEC 8802-2 LLC Type 2 setup		**0...n**	**class_id = 58, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. transmit_window_size_k	(static)	unsigned	1	127	1	x + 0x08
3. receive_window_size_rw	(static)	unsigned	1	127	1	x + 0x10
4. max_octets_i_pdu_n1	(static)	long unsigned			128	x + 0x18
5. max_number_transmissions_n2	(static)	unsigned				x + 0x20
6. acknowledgement_timer	(static)	long-unsigned				x + 0x28
7. p_bit_timer	(static)	long-unsigned				x + 0x30
8. reject_timer	(static)	long-unsigned				x + 0x38
9. busy_state_timer	(static)	long-unsigned				x + 0x40
Méthodes spécifiques		***o/f***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.11.3.2 Description d'attribut

4.11.3.2.1 logical_name

Identifie l'instance de l'objet "ISO/IEC 8802-2 LLC Type 2 setup". Voir 6.2.26.

transmit_window_size_k

La taille de fenêtre d'émission (k) doit être un paramètre de connexion de liaison de données qui ne peut jamais dépasser 127. Elle doit désigner le nombre maximal d'I PDU numérotées séquentiellement que la LLC émettrice peut avoir en cours (c'est-à-dire non acquittées). La valeur de k est le nombre maximal pour lequel la variable d'état V(S) d'envoi de LLC expéditrice peut dépasser le N(R) de la dernière I PDU reçue.

Voir ISO/IEC 8802-2:1998, 7.8.4, *Taille de fenêtre d'émission, k.*

4.11.3.2.2 receive_window_size_rw

La taille de fenêtre de réception (RW) doit être un paramètre de connexion de liaison de données qui ne peut jamais dépasser 127. Elle doit désigner le nombre maximal d'I PDU non acquittées numérotées séquentiellement que la LLC locale autorise la LLC distante à avoir en cours. Elle est émise dans le champ d'informations de XID (voir ISO/IEC 8802-2:1998, 5.4.1.1.2) et s'applique à l'expéditeur XID. Le destinataire XID doit mettre sa fenêtre d'émission (k) à une valeur inférieure ou égale à la fenêtre de réception de l'expéditeur XID afin d'éviter d'engorger l'expéditeur XID.

Voir ISO/IEC 8802-2:1998, 7.8.6, *Taille de fenêtre de réception, RW*.

4.11.3.2.3 max_octets_i_pdu_n1

N1 est un paramètre de connexion de liaison de données qui désigne le nombre maximal d'octets dans une I PDU. Se référer aux diverses descriptions MAC pour déterminer la valeur précise de N1 pour une méthode donnée d'accès au support. La LLC elle-même n'impose pas de restrictions sur la valeur de N1. Cependant, dans l'intérêt de disposer d'une valeur de N1 sur laquelle tous les utilisateurs de LLC de type 2 peuvent s'appuyer, tous les MAC doivent au moins être capables de prendre en charge les I PDU avec des champs d'informations d'une longueur maximale pouvant atteindre 128 octets.

Voir ISO/IEC 8802-2:1998, 7.8.3, *Nombre maximal d'octets dans I PDU, N1.*

4.11.3.2.4 max_number_transmissions_n2

N2 est un paramètre de connexion de liaison de données qui indique le nombre maximal de fois qu'une PDU est envoyée après l'expiration du temporisateur d'acquittement, du temporisateur de bit P, du temporisateur de rejet ou du temporisateur d'état occupé.

Voir ISO/IEC 8802-2:1998, 7.8.2, *Nombre maximal d'émissions, N2.*

4.11.3.2.5 acknowledgement_timer

Le temporisateur d'acquittement est un paramètre de connexion de liaison de données qui doit définir l'intervalle de temps pendant lequel la LLC doit s'attendre à recevoir un acquittement à une ou plusieurs I PDU ou une PDU de réponse attendue à une PDU de commande non numérotée envoyée. L'unité est la seconde.

Voir ISO/IEC 8802-2:1998, 7.8.1.1, *Temporisateur d'acquittement.*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.11.3.2.6 p_bit_timer

Le temporisateur de bit P est un paramètre de connexion de liaison de données qui doit définir l'intervalle de temps pendant lequel la LLC doit s'attendre à recevoir une PDU avec le bit F défini sur "1" en réponse à une commande de type 2 avec le bit P défini sur "1". L'unité est la seconde.

Voir ISO/IEC 8802-2:1998, 7.8.1.2, *Temporisateur de bit P.*

4.11.3.2.7 reject_timer

Le temporisateur de rejet est un paramètre de connexion de liaison de données qui doit définir l'intervalle de temps pendant lequel la LLC doit s'attendre à recevoir une réponse à une REJ PDU envoyée. L'unité est la seconde.

Voir ISO/IEC 8802-2:1998, 7.8.1.3, *Temporisateur de rejet.*

4.11.3.2.8 busy_state_timer

Le temporisateur d'état occupé est un paramètre de connexion de liaison de données qui doit définir l'intervalle de temps pendant lequel la LLC doit attendre une indication de la suppression d'un état occupé au niveau d'une autre LLC. L'unité est la seconde.

Voir ISO/IEC 8802-2:1998, 7.8.1.4, *Temporisateur d'état occupé*.

4.11.4 ISO/IEC 8802-2 LLC Type 3 setup (class_id = 59, version = 0)

4.11.4.1 Vue d'ensemble

Une instance de l'IC "ISO/IEC 8802-2 LLC Type 3 setup" contient les paramètres nécessaires pour établir la couche LLC selon l'ISO/IEC 8802-2 dans le fonctionnement de type 3.

ISO/IEC 8802-2 LLC Type 3 setup		**0…n**	**class_id = 59, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				X
2. max_octets_acn_pdu_n3	(static)	long unsigned				x + 0x08
3. max_number_transmissions_n4	(static)	unsigned				x + 0x10
4. acknowledgement_time_t1	(static)	long unsigned				x + 0x18
5. receive_lifetime_var_t2	(static)	long unsigned				x + 0x20
6. transmit_lifetime_var_t3	(static)	long unsigned				x + 0x28
Méthodes spécifiques		***o/f***				

4.11.4.2 Description d'attribut

4.11.4.2.1 logical_name

Identifie l'instance de l'objet "ISO/IEC 8802-2 LLC Type 3 setup". Voir 6.2.26.

4.11.4.2.2 max_octets_acn_pdu_n3

N3 est un paramètre de liaison logique qui désigne le nombre maximal d'octets dans une PDU de commande ACn. Se référer aux diverses descriptions MAC pour déterminer la valeur précise de N3 pour une méthode donnée d'accès au support. La LLC n'impose pas de restrictions sur la valeur de N3.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Voir ISO/IEC 8802-2:1998, 8.6.2, *Nombre maximal d'octets dans une PDU de commande ACn, N3.*

4.11.4.2.3 max_number_transmissions_n4

N4 est un paramètre de liaison logique qui indique le nombre maximal de fois qu'une PDU de commande ACn est envoyée par la LLC tentant de procéder à un échange d'informations réussi. Normalement, la valeur de N4 est suffisamment grande pour surmonter la perte d'une PDU due à des états d'erreur de liaison. Si la sous-couche de contrôle d'accès au support a sa propre capacité de retransmission, la valeur de N4 peut être définie sur 1 afin que la LLC ne remette pas elle-même en file d'attente une PDU destinée à la sous-couche de contrôle d'accès au support.

Voir ISO/IEC 8802-2:1998, 8.6.1, *Nombre maximal d'émissions, N4.*

4.11.4.2.4 acknowledgement_time_t1

Le temps d'acquittement est un paramètre de liaison logique qui détermine la période des temporisateurs d'acquittement et, à ce titre, doit définir l'intervalle de temps pendant lequel la LLC doit s'attendre à recevoir une PDU de réponse ACn de la part d'une LLC spécifique dont la LLC attend une PDU de réponse. Le temps d'acquittement doit prendre en compte tout retard introduit par la sous-couche MAC et le fait de savoir si le temporisateur a démarré au début ou à la fin de l'envoi de la PDU de commande ACn par la LLC. Le fonctionnement correct de la procédure doit exiger que le temps d'acquittement soit supérieur au temps normal entre l'envoi d'une PDU de commande ACn et la réception de la PDU de réponse ACn correspondante. Si la sous-couche de contrôle d'accès au support effectue ses propres retransmissions et si le paramètre de liaison logique N4 est défini sur 1 pour empêcher que la LLC ne remette en file d'attente une PDU, le temps d'acquittement T1 peut être mis à l'infini, rendant ainsi inutiles les temporisateurs d'acquittement.

L'unité est la seconde. L'infini est indiqué par le fait que tous les bits sont mis à 1.

Voir ISO/IEC 8802-2:1998, 8.6.4, *Temps d'acquittement, T1.*

4.11.4.2.5 receive_lifetime_var_t2

Cette valeur de temps est un paramètre de liaison logique qui détermine la période de tous les temporisateurs de durée de vie des variables de réception. T2 doit être plus long d'une certaine marge de sécurité que la période la plus longue possible pendant laquelle la première émission et toutes les nouvelles tentatives d'une simple PDU peuvent se produire. La marge de sécurité doit tenir compte de tout ce qui a une incidence sur la perception des LLC relatives à l'heure d'arrivée des PDU, telle que l'heure de réponse de LLC, la résolution du temporisateur et les variations temporelles exigées pour que la sous-couche de contrôle d'accès au support transmette les PDU reçues à la LLC.

Si la destruction des variables d'état reçues n'est pas souhaitée, la valeur du temps T2 peut être mise à l'infini. Dans ce cas, il n'est pas nécessaire de mettre en œuvre le temporisateur de durée de vie de variable de réception.

L'unité est la seconde. L'infini est indiqué par le fait que tous les bits sont mis à 1.

Voir ISO/IEC 8802-2:1998, 8.6.5, *Variable de durée de vie de réception, T2.*

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.11.4.2.6 transmit_lifetime_var_t3

Cette valeur de temps est un paramètre de liaison logique qui détermine la durée de vie minimale des variables d'état de séquence d'émission. T3 doit être plus long d'une certaine marge de sécurité que:

a) la variable de liaison logique T2 au niveau des stations vers lesquelles les commandes ACn sont envoyées;

b) la durée de vie la plus longue possible d'une paire commande-réponse ACn. La durée de vie d'une paire commande-réponse ACn doit prendre en compte la somme du temps de traitement, des retards de mise en file d'attente et du temps d'émission pour les PDU de commande et de réponse au niveau des stations locales et distantes.

Si la destruction des variables d'état d'émission n'est pas souhaitée, la valeur du temps T3 peut être mise à l'infini. A noter que, si le paramètre de durée de vie de variable de réception T2 est mis à l'infini sur les stations distantes vers lesquelles les commandes ACn sont envoyées, le paramètre T3 doit être mis à l'infini au niveau de la station locale.

L'unité est la seconde. L'infini est indiqué par le fait que tous les bits sont mis à 1.

Voir ISO/IEC 8802-2:1998, 8.6.6, *Variable de durée de vie d'émission, T3.*

4.12 Classes d'interfaces pour l'établissement et la gestion du profil PLC OFDM à bande étroite DLMS®/COSEM pour les réseaux PRIME

4.12.1 Vue d'ensemble

Voir aussi Annexe D.

Les objets COSEM pour l'échange de données utilisant le profil PLC OFDM à bande étroite pour les réseaux PRIME, s'il est mis en œuvre, doivent se trouver dans le dispositif logique de gestion des serveurs COSEM.

La Figure 28 donne un exemple avec un dispositif physique COSEM comprenant trois dispositifs logiques.

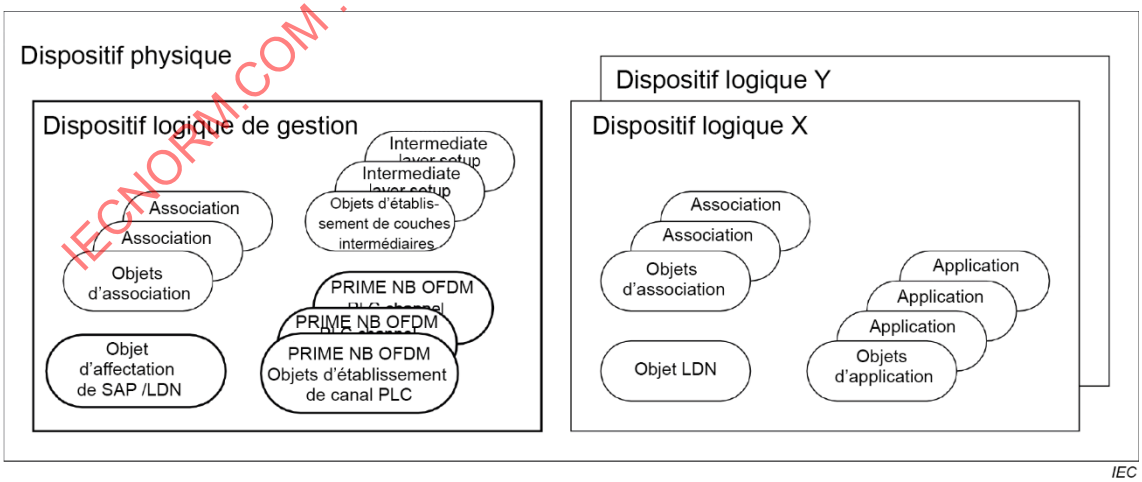


Figure 28 – Modèle d'objet des serveurs DLMS®/COSEM

Chaque dispositif logique doit contenir un objet LDN (Nom de dispositif logique).

NOTE Comme dans cet exemple, il y a plusieurs dispositifs logiques, le dispositif logique de gestion obligatoire contient un objet "SAP Assignment" en lieu et place d'un objet "LDN".

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Chaque dispositif logique contient un ou plusieurs objets "Association", un par client pris en charge.

Le dispositif logique de gestion contient les objets d'établissement des couches physiques et MAC du profil PLC OFDM à bande étroite pour les réseaux PRIME, ainsi que des objets d'établissement pour la ou les couches intermédiaires. Il peut contenir des objets d'applications supplémentaires.

Les autres dispositifs logiques contiennent, en plus des objets "Association" et "Logical Device Name" susmentionnés, d'autres objets d'applications, contenant des paramètres et des valeurs de mesure.

Pour configurer et gérer 61334-4-32 LLC SSCS, une IC est spécifiée:

- "61334-4-32 LLC SSCS setup" (voir 4.12.3).

Pour gérer la couche physique (PhL) PRIME NB OFDM PLC, une IC est spécifiée:

- "PRIME NB OFDM PLC Physical layer counters" (voir 4.12.5).

Pour configurer et gérer la couche PLC PRIME OFDM MAC, quatre IC sont spécifiées:

- "PRIME NB OFDM PLC MAC setup" (voir 4.12.6);
- "PRIME NB OFDM PLC MAC functional parameters" (voir 4.12.7);
- "PRIME NB OFDM PLC MAC counters" (voir 4.12.8);
- "PRIME NB OFDM PLC MAC network administration data" (voir 4.12.9).

Pour l'identification d'application, une IC est spécifiée:

- "PRIME NB OFDM PLC Application identification" (voir 4.12.11).

4.12.2 Mise en correspondance des attributs PRIME NB OFDM PLC PIB avec les attributs d'IC COSEM

L'UIT-T G.9904:2012 définit les variables des attributs PHY PIB (Tableau 10-1 et Tableau 10-2), les attributs MAC PIB (Tableau 10-3 au Tableau 10-8) et les attributs PIB d'application (Tableau 10-9).

Le Tableau 40 présente la mise en correspondance des attributs PRIME NB OFDM PLC PIB avec les attributs des IC COSEM. Seules les variables liées au commutateur et aux nœuds de terminaux sont mises en correspondance. Les variables pertinentes pour le nœud de base ne sont pas mises en correspondance, car le nœud de base agit comme un client pour le réseau de distribution.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Tableau 40 – Mise en correspondance des attributs PRIME NB OFDM PLC PIB avec les attributs d'IC COSEM

Nom	Identifiant	Classe d'interfaces	class_id/attribut
Attributs PHY PIB – Variable en lecture seule PHY donnant des informations statistiques [1]			
phyStatsCRCIncorrectCount	0x00A0	PRIME NB OFDM PLC Physical layer counters (class_id = 81, version = 0)	81 / Attr. 2
PhyStatsCRCFailCount	0x00A1		81 / Attr. 3
phyStatsTxDropCount	0x00A2		81 / Attr. 4
phyStatsRxDropCount	0x00A3		81 / Attr. 5
phyStatsRxTotalCount	0x00A4		Non modélisée
phyStatsBlkAvgEvm	0x00A5		Non modélisée
phyEmaSmoothing	0x00A8		Non modélisée
Paramètres en lecture seule PHY, donnant des informations relatives à la mise en œuvre spécifique [2]			
phyTxQueueLen	0x00B0		Non modélisée
phyRxQueueLen	0x00B1		
phyTxProcessingDelay	0x00B2		
phyRxProcessingDelay	0x00B3		
PhyAgcMinGain	0x00B4		
PhyAgcStepValue	0x00B5		
PhyAgcStepNumber	0x00B6		
Variables en lecture-écriture MAC, variables en lecture seule [3]			
macMinSwitchSearchTime	0x0010	PRIME NB OFDM PLC MAC setup (class_id = 82, version = 0)	82 / Attr. 2
macMaxPromotionPdu	0x0011		82 / Attr. 3
macMaxPromotionPduTxPeriod	0x0012		82 / Attr. 4
macBeaconsPerFrame	0x0013		82 / Attr. 5
macSCPMaxTxAttempts	0x0014		82 / Attr. 6
macCtlReTxTimer	0x0015		82 / Attr. 7
macMaxCtlReTx	0x0018		82 / Attr. 8
macEMASmoothing	0x0019		Non modélisée
macSCPRBO	0x0016		Non modélisée
macSCPChSenseCount	0x0017		Non modélisée
Variables en lecture seule MAC, donnant des informations fonctionnelles [4]			
macLNID	0x0020	NB OFDM PLC MAC functional parameters (class_id = 83 version = 0)	83 / Attr. 2
macLSID	0x0021		83 / Attr. 3
macSID	0x0022		83 / Attr. 4
macSNA	0x0023		83 / Attr. 5
macState	0x0024		83 / Attr. 6
macSCPLength	0x0025		83 / Attr. 7
macNodeHierarchyLevel	0x0026		83 / Attr. 8
macBeaconSlotCount	0x0027		83 / Attr. 9
macBeaconRxSlot	0x0028		83 / Attr. 10
macBeaconTxSlot	0x0029		83 / Attr. 11
macBeaconRxFrequency	0x002A		83 / Attr. 12
macBeaconTxFrequency	0x002B		83 / Attr. 13
macCapabilities	0x002C		83/ Attr. 14

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Nom	Identifiant	Classe d'interfaces	class_id/attribut
Variables en lecture seule MAC, donnant des informations statistiques [5]			
macTxDataPktCount	0x0040	PRIME NB OFDM PLC MAC counters (class_id = 84, version = 0)	84 / Attr. 2
macRxDataPktCount	0x0041		84 / Attr. 3
macTxCtrlPktCount	0x0042		84 / Attr. 4
macRxCtrlPktCount	0x0043		84 / Attr. 5
macCSMAFailCount	0x0044		84 / Attr. 6
macCSMAChBusyCount	0x0045		84 / Attr. 7
Listes en lecture seule, que la couche MAC a mises à disposition par l'intermédiaire de l'interface de gestion [6]			
macListMcastEntries	0x0052	PRIME NB OFDM PLC MAC network administration data (class_id = 85, version = 0)	85 / Attr. 2
macListSwitchTable	0x0053		85 / Attr. 3
macListDirectTable	0x0055		85 / Attr. 4
macListAvailableSwitches	0x0056		85 / Attr. 5
macListPhyComm	0x0057		85 / Attr. 6
Attributs PIB d'application [7]			
AppFwVersion	0x0075	PRIME NB OFDM PLC Application identification (class_id = 86, version = 0)	86 / Attr. 2
AppVendorId	0x0076		86 / Attr. 3
AppProductId	0x0077		86 / Attr. 4

[1] Voir UIT-T G.9904:2012, Tableau 10-1.

[2] Voir UIT-T G.9904:2012, Tableau 10-2.

[3] Voir UIT-T G.9904:2012, Tableaux 10-3, 10-4.

[4] Voir UIT-T G.9904:2012, Tableau 10-5.

[5] Voir UIT-T G.9904:2012, Tableau 10-6.

[6] Voir UIT-T G.9904:2012, Tableau 10-7.

[7] Voir UIT-T G.9904:2012, Tableau 10-9.

NOTE Dans les spécifications des classes d'interfaces COSEM, le trait de soulignement est utilisé, alors que dans la Recommandation UIT-T G.9904:2012 et dans le Tableau 1, la notation camel est utilisée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.3 61334-4-32 LLC SSCS setup (class_id = 80, version = 0)

4.12.3.1 Vue d'ensemble

Une instance de l'IC "61334-4-32 LLC SSCS" (Service Specific Convergence Sublayer, 432 CL) contient les adresses fournies par le nœud de base à l'ouverture de la couche de convergence, en réponse à la demande d'établissement du nœud de service. Elles permettent au nœud de service de faire partie intégrante du réseau géré par le nœud de base.

61334-4-32 LLC SSCS setup	**0...n**	**class_id = 80, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. service_node_address (dyn.)	long-unsigned				x + 0x08
3. base_node_address (dyn.)	long-unsigned				x + 0x10
Méthodes spécifiques	***o/f***				
1. reset (data)	f				x + 0x20

4.12.3.2 Description d'attribut

4.12.3.2.1 logical_name

Identifie l'instance de l'objet "61334-4-32 LLC SSCS setup". Voir 6.2.27.

4.12.3.2.2 service_node_address

Contient la valeur de l'adresse attribuée au nœud de service lors de son enregistrement par le nœud de base.

Après l'annulation de l'enregistrement, la valeur de cette adresse est NEW, soit 0xFFE.

4.12.3.2.3 base_node_address

Contient la valeur de l'adresse du nœud de base à laquelle est enregistré le nœud de service.

Après l'annulation de l'enregistrement, la valeur de cette adresse est 0.

4.12.3.3 Description de la méthode

4.12.3.3.1 reset (data)

Cette méthode est utilisée pour annuler l'attribution de l'adresse du nœud de service.

La valeur de *service_node_address* devient NEW, et celle de *base_node_address* devient 0.

4.12.4 Paramètres de couche physique PRIME NB OFDM PLC

Les paramètres de couche physique ne sont pas modélisés.

4.12.5 PRIME NB OFDM PLC Physical layer counters (class_id = 81, version = 0)

4.12.5.1 Vue d'ensemble

Une instance de l'IC "PRIME NB OFDM PLC Physical layer counters" enregistre les compteurs liés aux échanges des couches physiques. L'objectif de ces compteurs est de fournir des informations statistiques à des fins de gestion.

Les attributs des instances de cette IC doivent être en lecture seule. Ils peuvent être réinitialisés à l'aide de la méthode reset.

PRIME NB OFDM PLC Physical layer counters		**0…n**	**class_id = 81, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. phy_stats_crc_incorrect_count	(dyn.)	long-unsigned				x + 0x08
3. phy_stats_crc_fail_count	(dyn.)	long-unsigned				x + 0x10
4. phy_stats_tx_drop_count	(dyn.)	long-unsigned				x + 0x18
5. phy_stats_rx_drop_count	(dyn.)	long-unsigned				x + 0x20
Méthodes spécifiques		***o/f***				
1. reset (data)		f				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.5.2 Description d'attribut

4.12.5.2.1 Généralités

logical_name

Identifie l'instance d'objet "PRIME NB OFDM PLC Physical layer counters". Voir 6.2.27.

4.12.5.2.2 phy_stats_crc_incorrect_count

Attribut PIB 0x00A0: nombre de salves reçues sur la couche physique pour lesquelles le CRC était incorrect.

4.12.5.2.3 phy_stats_crc_failed_count

Attribut PIB 0x00A1: nombre de salves reçues sur la couche physique pour lesquelles le CRC était correct, mais dont le champ Protocol de l'en-tête PHY contenait une valeur non valide. Il s'agit du nombre de réceptions de données corrompues que le CRC n'a pas détectées.

4.12.5.2.4 phy_stats_tx_drop_count

Attribut PIB 0x00A2: nombre de fois où la couche PHY a reçu de nouvelles données à transmettre (PHY_DATA.request) et a dû écraser des données existantes dans sa file d'attente d'émission ou supprimer les données dans une nouvelle requête, car la file d'attente était saturée.

4.12.5.2.5 phy_stats_rx_drop_count

Attribut PIB 0x00A3: nombre de fois où la couche PHY a reçu de nouvelles données sur le canal et a dû écraser des données existantes dans sa file d'attente de réception ou supprimer les données qu'elle venait de recevoir, car la file d'attente était saturée.

NOTE Si un compteur atteint la valeur maximale (0xFFFF), il est automatiquement arrêté.

4.12.5.3 Description de la méthode

4.12.5.3.1 reset (data)

Cette méthode est utilisée pour réinitialiser tous les compteurs d'une instance de cette interface.

4.12.6 PRIME NB OFDM PLC MAC setup (class_id = 82, version = 0)

4.12.6.1 Vue d'ensemble

Une instance de l'IC "PRIME NB OFDM PLC MAC setup" contient les paramètres nécessaires à l'établissement et la gestion de la couche PRIME NB OFDM PLC MAC.

Ces attributs influent sur le comportement fonctionnel d'une mise en œuvre. Ces attributs peuvent être définis hors de la couche MAC, en général par l'entité de gestion, et les mises en œuvre peuvent permettre de modifier leurs valeurs pendant le fonctionnement normal, c'est-à-dire même après la séquence de démarrage du dispositif.

NOTE Si un compteur atteint la valeur maximale (0xFFFF), il est automatiquement arrêté.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PRIME NB OFDM PLC MAC setup		0...n	class_id = 82, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mac_min_switch_search_time	(static)	unsigned	16	32	24	x + 0x08
3. mac_max_promotion_pdu	(static)	unsigned	1	4	2	x + 0x10
4. mac_promotion_pdu_tx_period	(static)	unsigned	2	8	5	x + 0x18
5. mac_beacons_per_frame	(static)	unsigned	1	5	5	x + 0x20
6. mac_scp_max_tx_attempts	(static)	unsigned	2	5	5	x +0x28
7. mac_ctl_re_tx_timer	(static)	unsigned	2	20	15	x + 0x30
8. mac_max_ctl_re_tx	(static)	unsigned	3	5	3	x + 0x38
Méthodes spécifiques		***o/f***				

4.12.6.2 Description d'attribut

4.12.6.2.1 logical_name

Identifie l'instance de l'objet "PRIME NB OFDM PLC MAC setup". Voir 6.2.27.

4.12.6.2.2 mac_min_switch_search_time

Attribut PIB 0x0010: durée minimale pendant laquelle il convient qu'un nœud de service à l'état *Disconnected* analyse le canal à la recherche d'étiquettes avant de pouvoir diffuser une PNPDU. Cet attribut n'est pas conservé dans les nœuds de base.

L'unité de cet attribut est la seconde.

4.12.6.2.3 mac_max_promotion_pdu

Attribut PIB 0x0011: nombre maximal de PNPDU qui peuvent être transmises par un nœud de service dans une période de *mac_promotion_pdu_tx_period* secondes. Cet attribut n'est pas conservé dans les nœuds de base.

4.12.6.2.4 mac_promotion_pdu_tx_period

Attribut PIB 0x0012: intervalle de temps de limitation du nombre de PNDPU transmises à partir d'un nœud de service. Pas plus que *mac_max_promotion_pdu* ne peuvent être transmises sur une période *mac_promotion_pdu_tx_period*.

L'unité de cet attribut est la seconde.

4.12.6.2.5 mac_beacons_per_frame

Attribut PIB 0x0013: nombre maximal d'emplacements d'étiquette qui peuvent être prévus dans une trame. Cet attribut est conservé dans les nœuds de base.

4.12.6.2.6 mac_scp_max_tx_attempts

Attribut PIB 0x0014: nombre de fois où l'algorithme CSMA tente de transmettre les données demandées lorsqu'une tentative précédente a été différée, la couche PHY ayant indiqué un canal occupé.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.6.2.7 mac_ctl_re_tx_timer

Attribut PIB 0x0015: nombre de secondes pendant lesquelles une entité MAC attend l'acquittement de la réception du paquet de commande MAC de la part de son entité homologue. À l'expiration de cette période, l'entité MAC peut retransmettre le paquet de commande MAC.

L'unité de cet attribut est la seconde.

4.12.6.2.8 mac_max_ctl_re_tx

Attribut PIB 0x0018: nombre maximal de fois où une entité MAC tente de retransmettre un paquet de commande MAC qui n'a pas été acquitté. Si le nombre de retransmissions atteint la valeur maximale, l'entité MAC doit annuler les tentatives supplémentaires de retransmission du paquet de commande MAC.

NOTE Si un compteur atteint la valeur maximale (0xFFFF), il est automatiquement arrêté.

4.12.7 NB OFDM PLC MAC functional parameters (class_id = 83 version = 0)

4.12.7.1 Vue d'ensemble

Les attributs d'une instance de l'IC "PRIME NB OFDM PLC MAC functional parameters" appartiennent au comportement fonctionnel de la couche MAC. Ils donnent des informations relatives à des aspects spécifiques.

Les attributs des instances de cette IC doivent être en lecture seule.

PRIME NB OFDM PLC MAC functional parameters		**0...n**	**class_id = 83, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mac_LNID	(static)	long	0	16 383		x + 0x08
3. mac_LSID	(static)	unsigned	0	255		x + 0x10
4. mac_SID	(static)	unsigned	0	255		x + 0x18
5. mac_SNA	(static)	octet-string				x + 0x20
6. mac_state	(static)	enum	0	3		x +0x28
7. mac_scp_length	(static)	long				x + 0x30
8. mac_node_hierarchy_level	(static)	unsigned	0	63		x + 0x38
9. mac_beacon_slot_count	(static)	unsigned	0	7		x + 0x40
10. mac_beacon_rx_slot	(static)	unsigned	0	7		x + 0x48
11. mac_beacon_tx_slot	(static)	unsigned	0	7		x + 0x50
12. mac_beacon_rx_frequency	(static)	unsigned	0	31		x + 0x58
13. mac_beacon_tx_frequency	(static)	unsigned	0	31		x + 0x60
14. mac_capabilities	(static)	long-unsigned				x + 0x68
Méthodes spécifiques		***o/f***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.7.2 Description d'attribut

4.12.7.2.1 logical_name

Identifie l'instance de l'objet "PRIME NB OFDM PLC MAC functional parameters". Voir 6.2.27.

4.12.7.2.2 mac_LNID

Attribut PIB 0x0020: LNID attribué à ce nœud au moment de son enregistrement.

4.12.7.2.3 mac_LSID

Attribut PIB 0x0021: LSID attribué à ce nœud au moment de sa promotion. Cet attribut n'est pas conservé si le nœud est à l'état fonctionnel *Terminal*.

4.12.7.2.4 mac_SID

Attribut PIB 0x0022: SID du nœud de commutation via lequel ce nœud est connecté au sous-réseau. Cet attribut n'est pas conservé dans un nœud de base.

4.12.7.2.5 mac_SNA

Attribut PIB 0x0023: adresse de sous-réseau à laquelle ce nœud est enregistré. Le nœud de base retourne le SNA qu'il utilise.

4.12.7.2.6 mac_state

Attribut PIB 0x0024: état fonctionnel actuel du nœud.

enum:

(0) Disconnected,
(1) Terminal,
(2) Switch,
(3) Base

4.12.7.2.7 mac_scp_length

Attribut PIB 0x0025: longueur de SCP, en nombre de symboles, dans la trame en cours.

4.12.7.2.8 mac_node_hierarchy_level

Attribut PIB 0x0026: niveau de ce nœud dans la hiérarchie de sous-réseau.

4.12.7.2.9 mac_beacon_slot_count

Attribut PIB 0x0027: nombre d'emplacements d'étiquettes prévus dans la structure de trame en cours.

4.12.7.2.10 mac_beacon_rx_slot

Attribut PIB 0x0028: emplacement d'étiquette dans lequel le nœud de commutation de ce dispositif transmet son étiquette. Cet attribut n'est pas conservé dans un nœud de base.

4.12.7.2.11 mac_beacon_tx_slot

Attribut PIB 0x0029: emplacement d'étiquette dans lequel ce dispositif transmet son étiquette. Cet attribut n'est pas conservé dans les nœuds de service à l'état fonctionnel *Terminal*.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.7.2.12 mac_beacon_rx_frequency

Attribut PIB 0x002A: nombre de trames entre les réceptions de deux étiquettes successives. Une valeur 0x0 indique les étiquettes reçues dans chaque trame. Cet attribut n'est pas conservé dans un nœud de base.

4.12.7.2.13 mac_beacon_tx_frequency

Attribut PIB 0x002B: nombre de trames entre les transmissions de deux étiquettes successives. Une valeur 0x0 indique les étiquettes transmises dans chaque trame. Cet attribut n'est pas conservé dans les nœuds de service à l'état fonctionnel *Terminal*.

4.12.7.2.14 mac_capabilities

Attribut PIB 0x002C: cet attribut définit les capacités du nœud. Il s'agit d'une carte binaire qui définit une capacité.

Bit 0: capacité de commutation
Bit 1: agrégation de paquets
Bit 2: période sans conflit
Bit 3: connexion directe
Bit 4: multicast
Bit 5: gestion de solidité PHY
Bit 6: ARQ
Bit 7: réservé pour utilisation ultérieure
Bit 8: commutation de connexion directe
Bit 9: capacité de commutation multicast
Bit 10: capacité de commutation de gestion de solidité PHY
Bit 11: capacité de commutation de mise en mémoire tampon ARQ
Bits 12 à 15: réservés pour utilisation ultérieure

4.12.8 PRIME NB OFDM PLC MAC counters (class_id = 84, version = 0)

4.12.8.1 Vue d'ensemble

Une instance de l'IC "PRIME NB OFDM PLC MAC counters" enregistre des informations statistiques sur le fonctionnement de la couche MAC pour les besoins de la gestion. Les attributs des instances de cette IC doivent être en lecture seule. Ils peuvent être réinitialisés à l'aide de la méthode *reset*.

NOTE Si un compteur atteint la valeur maximale (0xFFFF), il est automatiquement arrêté.

PRIME NB OFDM PLC MAC counters	**0…n**	**class_id = 84, version = 0**			
Attributs	***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name (static)	octet-string				x
2. mac_tx_data_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x08
3. mac_rx_data_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x10
4. mac_tx_ctrl_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x18
5. mac_rx_ctrl_pkt_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x20
6. mac_csma_fail_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x28
7. mac_csma_ch_busy_count (dyn.)	double-long-unsigned		4 294 967 295		x + 0x30
Méthodes spécifiques	***o/f***				
1. reset (data)	f				x + 0x40

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.8.2 Description d'attribut

4.12.8.2.1 logical_name

Identifie l'instance de l'objet "PRIME NB OFDM PLC MAC counters". Voir 6.2.27.

4.12.8.2.2 mac_tx_data_pkt_count

Attribut PIB 0x0040: compte les MSDU dont la transmission a abouti.

4.12.8.2.3 mac_rx_data_pkt_count

Attribut PIB 0x0041: compte les MSDU reçues dont l'adresse de destination était ce nœud.

4.12.8.2.4 mac_tx_ctrl_pkt_count

Attribut PIB 0x0042: compte les paquets de commande MAC dont la transmission a abouti.

4.12.8.2.5 mac_rx_ctrl_pkt_count

Attribut PIB 0x0043: compte les paquets de commande MAC reçus dont la destination était ce nœud.

4.12.8.2.6 mac_csma_fail_count

Attribut PIB 0x0044: compte les tentatives de transmission CSMA qui n'ont pas abouti.

4.12.8.2.7 mac_csma_ch_busy_count

Attribut PIB 0x0045: compte le nombre de fois où ce nœud a dû réduire la transmission SCP en raison de l'état occupé du canal.

NOTE Si un compteur atteint la valeur maximale (0xFFFFFFFF), il est automatiquement arrêté.

4.12.8.3 Description de la méthode

4.12.8.3.1 reset (data)

Cette méthode est utilisée pour réinitialiser tous les compteurs d'une instance de cette classe d'interfaces.

data::= integer (0)

4.12.9 PRIME NB OFDM PLC MAC network administration data (class_id = 85, version = 0)

4.12.9.1 Vue d'ensemble

Cette IC contient les paramètres liés à la gestion des dispositifs connectés au réseau.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PRIME NB OFDM PLC MAC network administration data		**0...n**	**class_id = 85, version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mac_list_multicast_entries	(dyn.)	array				x + 0x08
3. mac_list_switch_table	(dyn.)	array				x + 0x10
4. mac_list_direct_table	(dyn.)	array				x + 0x18
5. mac_list_available_switches	(dyn.)	array				x + 0x20
6. mac_list_phy_comm	(dyn)	array				x + 0x28
Méthodes spécifiques		***o/f***				
1. reset (data)		*f*				x + 0x30

4.12.9.2 Description d'attribut

4.12.9.2.1 logical_name

Identifie l'instance de l'objet "PRIME NB OFDM PLC MAC network administration data". Voir 6.2.27.

4.12.9.2.2 mac_list_multicast_entries

Attribut PIB 0x0052: liste des entrées dans la table de commutation multicast. Cette liste n'est pas conservée dans les nœuds de service à l'état fonctionnel *Terminal*.

```
mac_list_multicast_entries_type::= array mac_list_multicast_entries_element

mac_list_multicast_entries_element::= structure

{
mcast_entry_LCID:          integer, -- LCID du groupe multicast
mcast_entry_members:       long     -- nombre de nœuds enfants
}
```

Le nombre de nœuds enfants est le nombre de membres de ce groupe, y compris le nœud lui-même.

4.12.9.2.3 mac_list_switch_table

Attribut PIB 0x0053: tableau de commutation. Ce tableau n'est pas conservé par les nœuds de service à l'état *Terminal*.

```
mac switch_table::= array     stbl_entry_LSID

stbl_entry_LSID::=      long  -- SID du nœud de commutation connexe
```

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.12.9.2.4 mac_list_direct_table

Attribut PIB 0x0055: tableau direct.

```
mac_direct_table::= array     mac_direct_table_element

mac_direct_table_element::= structure
{
dconn_entry_src_SID:          long,
dconn_entry_src_LNID:         long,
dconn_entry_src_LCID:         long,
dconn_entry_dst_SID:          long,
dconn_entry_dst_LNID:         long,
dconn_entry_dst_LCID:         long,
dconn_entry_DID:              octet-string (6 octets)
}
```

Où:

– dconn_entry_src_SID est le SID du commutateur par lequel le nœud de service source est connecté;

– dconn_entry_src_LNID est le NID attribué au nœud de service source;

– dconn_entry_src_LCID est le LCID attribué à cette connexion au niveau de la source;

– dconn_entry_dst_SID est le SID du commutateur par lequel le nœud de service de destination est connecté;

– dconn_entry_dst_LNID est le NID attribué au nœud de service de destination;

– dconn_entry_dst_LCID est le LCID attribué à cette connexion au niveau de la destination;

– dconn_entry_DID est l'EUI-48 du commutateur direct.

4.12.9.2.5 mac_list_available_switches

Attribut PIB 0x0056: liste des nœuds de commutation dont les étiquettes sont reçues.

```
mac_list_available_switches::= array mac_list_available_switches_element

mac_list_available_switches_element::= structure

{
slist_entry_SNA:          octet-string (6 octets),
slist_entry_LSID:         long,
slist_entry_level:        integer,
slist_entry_rx_level:     integer,
slist_entry_rx_snr:       integer
}
```

Où:

– slist_entry_SNA est l'EUI-48 du sous-réseau;

– slist_entry_LSID est le SID de ce commutateur;

– slist_entry_level est le niveau de ce commutateur dans la hiérarchie de sous-réseau;

– slist_entry_rx_level est le niveau de signal reçu pour ce commutateur;

– slist_entry_rx_snr est le rapport signal-bruit pour ce commutateur.

4.12.9.2.6 mac_list_phy_comm

Attribut PIB 0x0057: liste des paramètres de communication PHY. Il est maintenu dans chaque nœud. Pour les nœuds de terminal, il contient une seule entrée pour le commutateur par lequel le nœud est connecté. Pour les autres nœuds, il contient également des entrées pour chaque nœud enfant directement connecté.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

```
mac_list_phy_comm::= array      phy_comm_element

phy_comm_element::= structure

{
phy_Comm_EUI:              octet-string,
phy_Comm_Tx_Pwr:           integer,
phy_Comm_Tx_Cod:           integer,
phy_Comm_Rx_Cod:           integer,
phy_Comm_Rx_Lvl:           integer,
phy_Comm_SNR:              integer,
phy_Comm_Tx_Pwr_Mod:       integer,
phy_Comm_Tx_Cod_Mod:       integer,
phy_Comm_Rx_Cod_Mod:       integer
}
```

Où:

– phy_Comm_EUI est l'EUI-48 de l'autre dispositif;

– phy_Comm_Tx_Pwr est la puissance de l'émetteur de paquets GPDU envoyés au dispositif;

– phy_Comm_Tx_Cod est le codage de l'émetteur de paquets GPDU envoyés au dispositif;

– phy_Comm_Rx_Cod est le codage du récepteur de paquets GPDU reçus du dispositif;

– phy_Comm_Rx_Lvl est le niveau de puissance du récepteur de paquets GPDU reçus du dispositif;

– phy_Comm_SNR est le SNR des paquets GPDU reçus du dispositif;

– phy_Comm_Tx_Pwr_Mod est le nombre de fois où la puissance de l'émetteur a été modifiée;

– phy_Comm_Tx_Cod_Mod est le nombre de fois où le codage de l'émetteur a été modifié;

– phy_Comm_Rx_Cod_Mod est le nombre de fois où le codage du récepteur a été modifié.

4.12.9.3 Description de la méthode

4.12.9.3.1 reset (data)

Cette méthode est utilisée pour redéfinir toutes les entrées (d'un tableau à 0 élément) des attributs 2 à 6 de l'instance de cette classe d'interfaces.

```
data:= integer (0)
```

4.12.10 PRIME NB OFDM PLC MAC address setup (class_id = 43, version = 0)

Une instance de l'IC "MAC address setup" contient l'adresse MAC EUI-48 du dispositif. La taille de cette chaîne d'octets est de 6, compte tenu du fait que cette adresse est un EUI-48 et qu'elle est unique. Voir aussi 4.9.4 et 6.2.27.

4.12.11 PRIME NB OFDM PLC Application identification (class_id = 86, version = 0)

4.12.11.1 Vue d'ensemble

Une instance de l'IC "PRIME NB OFDM PLC Applications identification" contient les informations d'identification relatives à l'administration et la maintenance des dispositifs PRIME NB OFDM PLC. Il ne s'agit pas de paramètres de communication, mais ils permettent néanmoins de gérer le dispositif.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

PRIME NB OFDM PLC Application identification		0...n	class_id = 86, version = 0			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. firmware_version	(static)	octet-string		128		x + 0x08
3. vendor_Id	(static)	long-unsigned				x + 0x10
4. product_Id	(static)	long-unsigned				x + 0x18
Méthodes spécifiques		***o/f***				

4.12.11.2 Description d'attribut

4.12.11.2.1 logical_name

Identifie l'instance de l'objet d'établissement de dispositif. Voir 6.2.27.

4.12.11.2.2 firmware_version

Attribut PIB 0x0075: description textuelle de la version du progiciel s'exécutant sur le dispositif.

4.12.11.2.3 vendor_Id

Attribut PIB 0x0076: identifiant unique du fournisseur attribué par PRIME Alliance.

4.12.11.2.4 product_Id

Attribut PIB 0x0077: identifiant unique attribué par le fournisseur pour un produit spécifique.

4.13 Classes d'interfaces pour l'établissement et la gestion du profil PLC OFDM à bande étroite DLMS®/COSEM pour les réseaux G3-PLC

4.13.1 Vue d'ensemble

Le présent paragraphe 4.13 spécifie la version 1 des classes d'interfaces pour l'établissement et la gestion des couches MAC et d'adaptation 6LoWPAN du profil DLMS®/COSEM G3-PLC, conformément à l'UIT-T G.9903 Amd. 1:2017. Voir aussi Annexe E.

NOTE 1 L'UIT-T G.9903:2013 a été prise en charge en utilisant la version 0 de ces classes d'interfaces qui sont conservées en 5.13.

NOTE 2 L'UIT-T G.9903:2014 a été prise en charge en utilisant la version 1 de l'IC "G3-PLC MAC layer counters" de la section 5.13.5, la version 1 de l'IC "G3-PLC MAC setup" qui est conservée en 5.13.6, et la version 1 de l'IC "G3-PLC 6LoWPAN adaptation layer setup" qui est conservée en 5.13.7.

À cet effet, les éléments de la base d'informations de PAN (PIB) ont été mis en correspondance avec les attributs des IC COSEM.

Les objets COSEM pour l'échange de données utilisant le profil G3-PLC, s'il est mis en œuvre, doivent se trouver dans le dispositif logique de gestion des serveurs COSEM.

Pour établir et gérer les couches de profil DLMS®/COSEM G3-PLC (y compris PHY, IEEE 802.15.4:2006, MAC et 6LoWPAN), trois IC sont spécifiées:

- "G3-PLC MAC layer counters" (voir 4.13.3);
- "G3-PLC MAC setup" (voir 4.13.4);
- "G3-PLC 6LoWPAN adaptation layer setup" (voir 4.13.5).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Une instance de la classe d'interfaces COSEM existante "MAC address" (class_id: 43, version: 0) est nécessaire pour indiquer l'adresse MAC EUI-48 du modem G3-PLC (correspondant à la constante aExtendedAddress dans l'IEEE 802.15.4:2006).

La configuration IPv6 est fournie par une instance de la classe "IPv6 setup".

NOTE 3 La couche PHY de l'UIT-T G.9903 Amd. 1:2017 est hors du domaine d'application des IC "G3-PLC setup".

4.13.2 Mise en correspondance des attributs G3-PLC PIB avec les attributs d'IC COSEM

Selon les termes de l'IEEE 802.15.4:2006, un compteur est un dispositif à fonction réduite (RFD) alors qu'un concentrateur/point d'accès aux réseaux de voisinage (NNAP) est un dispositif à fonction intégrale (FFD)/coordinateur PAN. Selon les termes du DLMS®/COSEM, le compteur est le serveur et le concentrateur/NNAP est le client (ou un agent pour le compte du client).

Le COSEM modélisant uniquement le serveur et non le client, les classes de configuration G3-PLC concernent uniquement le RFD (dispositif à fonction réduite) et non le coordinateur PAN.

Le Tableau 41 présente la mise en correspondance des attributs G3-PLC PIB avec les attributs des classes d'interfaces COSEM.

Tableau 41 – Mise en correspondance des attributs G3-PLC PIB avec les attributs d'IC COSEM

Nom	Identifiant	Classe d'interfaces	class_id/attribut
Compteurs MAC – Attributs PIB en lecture seule, donnant des informations statistiques [1]			
mac_Tx_data_packet_count	0x0101	G3-PLC MAC layer counters (class_id = 90, version = 1)	90 / Att. 2
mac_Rx_data_packet_count	0x0102		90 / Att. 3
mac_Tx_cmd_packet_count	0x0103		90 / Att. 4
mac_Rx_cmd_packet_count	0x0104		90 / Att. 5
mac_CSMA_fail_count	0x0105		90 / Att. 6
mac_CSMA_no_ACK_count	0x0106		90 / Att. 7
mac_bad_CRC_count	0x0109		90 / Att. 8
mac_Tx_data_broadcast_count	0x0108		90 / Att. 9
mac_Rx_data_broadcast_count	0x0107		90 / Att. 10
Attributs PIB de définition MAC – Variables en lecture seule, en lecture-écriture et en écriture seule [1,2]			
mac_short_address	0x0053	G3-PLC MAC setup (class_id = 91, version = 2)	91 / Att. 2
mac_RC_coord	0x010F		91 / Att. 3
mac_PAN_id	0x0050		91 / Att. 4
mac_key_table	0x0071		91 / Att. 5
mac_frame_counter	0x0077		91 / Att. 6
mac_tone_mask	0x0110		91 / Att. 7
mac_TMR_TTL	0x010D		91 / Att. 8
mac_max_frame_retries	0x0059		91 / Att. 9
mac_neighbour_table_entry_TTL	0x010E		91 / Att. 10
mac_neighbour_table	0x010A		91 / Att. 11
mac_high_priority_window_size	0x0100		91 / Att. 12
mac_CSMA_fairness_limit	0x010C		91 / Att. 13
mac_beacon_randomization_window_length	0x0111		91 / Att. 14

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Nom	Identifiant	Classe d'interfaces	class_id/attribut
mac_A	0x0112		91 / Att. 15
mac_K	0x0113		91 / Att. 16
mac_min_CW_attempts	0x0114		91 / Att. 17
mac_cenelec_legacy_mode	0x0115		91 / Att. 18
mac_FCC_legacy_mode	0x0116		91 / Att. 19
mac_max_BE	0x0047		91 / Att. 20
mac_max_CSMA_backoff	0x004E		91 / Att. 21
mac_min_BE	0x004F		91 / Att. 22
Attributs PIB de couche d'adaptation 6LoWPAN – Variables en lecture seule et en lecture-écriture [3, 4]			
adp_max_hops	0x0F	G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 2)	92 / Att. 2
adp_weak_LQI_value	0x1A		92 / Att. 3
adp_security_level	0x00		92 / Att. 4
adp_prefix_table	0x01		92 / Att. 5
adp_routing_configuration	0x09, 0x0A, 0x0D, 0x11-0x19, 0x1B, 0x1F		92 / Att. 6
adp_broadcast_log_table_entry_TTL	0x02		92 / Att. 7
adp_routing_table	0x0C		92 / Att. 8
adp_context_information_table	0x07		92 / Att. 9
adp_blacklist_table	0x1E		92 / Att. 10
adp_broadcast_log_table	0x0B		92 / Att. 11
adp_group_table	0x0E		92 / Att. 12
adp_max_join_wait_time	0x20		92 / Att. 13
adp_path_discovery_time	0x21		92 / Att. 14
adp_active_key_index	0x22		92 / Att. 15
adp_metric_type	0x03		92 / Att. 16
adp_coord_short_address	0x08		92 / Att. 17
adp_disable_default_routing	0xF0		92 / Att. 18
adp_device_type	0x10		92 / Att. 19

[1] Voir UIT-T G.9903 Amd. 1:2017, 9.3.6.2.2 et 9.3.6.2.3.

[2] Les attributs suivants des attributs PIB de la sous-couche MAC G3-PLC ont été exclus, car il n'était pas utile de les exposer: *macBSN, macDSN, macAckWaitDuration, macFreqNotching, macTimeStampSupported, macPromiscuousMode, macSecurityEnabled*.

[3] Voir UIT-T G.9903 Amd. 1:2017, 9.4.1.1.

[4] Les attributs suivants des attributs PIB de la sous-couche Adaptation G3-PLC ont été exclus, car il n'était pas utile de les exposer: *adpSoftVersion, adpSnifferMode*.

NOTE Alors que l'UIT-T G.9903 Amd. 1:2017 utilise la notation camel, les spécifications des classes d'interfaces COSEM (et le présent tableau) utilisent la notation à trait de soulignement.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.3 G3-PLC MAC layer counters (class_id = 90, version = 1)

4.13.3.1 Vue d'ensemble

Une instance de l'IC "G3-PLC MAC layer counters" enregistre les compteurs liés aux échanges de couche MAC. L'objectif de ces compteurs est de fournir des informations statistiques à des fins de gestion.

Les attributs des instances de cette IC doivent être en lecture seule. Ils peuvent être réinitialisés à l'aide de la méthode *reset*.

G3-PLC MAC layer counters		**0…n**	**class_id = 90, version = 1**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mac_Tx_data_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x08
3. mac_Rx_data_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x10
4. mac_Tx_cmd_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x18
5. mac_Rx_cmd_packet_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x20
6. mac_CSMA_fail_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x28
7. mac_CSMA_no_ACK_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x30
8. mac_bad_CRC_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x38
9. mac_Tx_data_broadcast_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x40
10. mac_Rx_data_broadcast_count	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x48
Méthodes spécifiques		***o/f***				
1. reset (data)		*f*				x + 0x50

4.13.3.2 Description d'attribut

NOTE Si un compteur atteint la valeur maximale (0xFFFFFFFF), il est automatiquement arrêté.

4.13.3.2.1 logical_name

Identifie l'instance de l'objet "G3-PLC MAC layer counters". Voir 6.2.28.

4.13.3.2.2 mac_Tx_data_packet_count

Attribut PIB 0x0101: compteur statistique des paquets de données dont la transmission a abouti (MSDU).

4.13.3.2.3 mac_Rx_data_packet_count

Attribut PIB 0x0102: compteur statistique des paquets de données dont la réception a abouti (MSDU).

4.13.3.2.4 mac_Tx_cmd_packet_count

Attribut PIB 0x0103: compteur statistique des paquets de commande dont la transmission a abouti.

4.13.3.2.5 mac_Rx_cmd_packet_count

Attribut PIB 0x0104: compteur statistique des paquets de commande dont la réception a abouti.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.3.2.6 mac_CSMA_fail_count

Attribut PIB 0x0105: compte le nombre de fois où des reculs CSMA atteignent macMaxCSMABackoffs.

4.13.3.2.7 mac_CSMA_no_ACK_count

Attribut PIB 0x0106: compte le nombre de fois où un acquittement (ACK) n'est pas reçu lors de la transmission d'une trame de données monodiffusion (la perte de l'ACK est attribuée à des collisions).

4.13.3.2.8 mac_bad_CRC_count

Attribut PIB 0x0109: compteur statistique du nombre de trames reçues avec un CRC incorrect.

4.13.3.2.9 mac_Tx_data_broadcast_count

Attribut PIB 0x0108: compteur statistique du nombre de trames de diffusion envoyées.

4.13.3.2.10 mac_Rx_data_broadcast_count

Attribut PIB 0x0107: compteur statistique des paquets de diffusion dont la réception a abouti.

4.13.3.3 Description de la méthode

4.13.3.3.1 reset (data)

Cette méthode force une réinitialisation de l'objet. En appelant cette méthode, la valeur 0 est attribuée à tous les compteurs.

data::= integer (0)

4.13.4 G3-PLC MAC setup (class_id = 91, version = 2)

4.13.4.1 Vue d'ensemble

Une instance de l'IC "G3-PLC MAC setup" contient les paramètres nécessaires à l'établissement et la gestion de la sous-couche MAC G3-PLC selon l'IEEE 802.15.4:2006.

Ces attributs influent sur le comportement fonctionnel d'une mise en œuvre. Les mises en œuvre peuvent permettre de modifier les attributs pendant le fonctionnement normal, c'est-à-dire même si la séquence de démarrage du dispositif a été exécutée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3-PLC MAC setup		0...n	class_id = 91, version = 1			
Attributs		**Type de données**	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name	(static)	octet-string				x
2. mac_short_address	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x08
3. mac_RC_coord	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x10
4. mac_PAN_id	(dyn.)	long-unsigned	0x0000	0xFFFF	0xFFFF	x + 0x18
5. mac_key_table	(dyn.)	array				x + 0x20
6. mac_frame_counter	(dyn.)	double-long-unsigned	0	4 294 967 295	0	x + 0x28
7. mac_tone_mask	(static)	bit-string			0x000000 0000FF FFFFFF F	x + 0x30
8. mac_TMR_TTL	(static)	unsigned	0	255	2	x + 0x38
9. mac_max_frame_retries	(static)	unsigned	0	10	5	x + 0x40
10. mac_neighbour_table_entry_TTL	(static)	unsigned	0	255	255	x + 0x48
11. mac_neighbour_table	(dyn.)	array				x + 0x50
12. mac_high_priority_window_size	(static)	unsigned	1	7	7	x + 0x58
13. mac_CSMA_fairness_limit	(static)	unsigned	Voir ci-dessous	255	25	x + 0x60
14. mac_beacon_randomization_ window_length	(static)	unsigned	1	254	12	x + 0x68
15. mac_A	(static)	unsigned	3	20	8	x + 0x70
16. mac_K	(static)	unsigned	1	Voir ci-dessous	5	x + 0x78
17. mac_min_CW_attempts	(static)	unsigned	0	255	10	x + 0x80
18. mac_cenelec_legacy_mode	(static)	unsigned	0	255	1	x + 0x88
19. mac_FCC_legacy_mode	(static)	unsigned	0	255	1	x + 0x90
20. mac_max_BE	(static)	unsigned	0	20	8	x + 0x98
21. mac_max_CSMA_backoff	(static)	unsigned	0	255	50	x + 0xA0
22. mac_min_BE	(static)	unsigned	0	20	3	x + 0xA8
Méthodes spécifiques		***o/f***				
1. mac_get_neighbour_table_entry (data)		f				x + 0xB0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2 Description d'attribut

4.13.4.2.1 logical_name

Identifie l'instance de l'objet "G3-PLC MAC setup". Voir 6.2.28.

4.13.4.2.2 mac_short_address

Attribut PIB 0x0053: adresse de 16 bits que le dispositif utilise pour communiquer avec le réseau PAN. Sa valeur doit être égale à 0xFFFF lorsque le dispositif ne dispose d'aucune adresse courte. Un dispositif associé dispose nécessairement d'une adresse courte, de sorte qu'il ne puisse pas être associé sans adresse courte.

4.13.4.2.3 mac_RC_coord

Attribut PIB 0x010F: Coût de liaison vers le coordinateur, à utiliser dans la charge utile d'étiquette en tant que RC_COORD

4.13.4.2.4 mac_PAN_id

Attribut PIB 0x0050: identifiant à 16 bits du réseau PAN sur lequel fonctionne le dispositif. Une valeur égale à 0xFFFF indique que le dispositif n'est pas associé.

4.13.4.2.5 mac_key_table

Attribut PIB 0x0071: cet attribut contient les clés GMK exigées pour le chiffrement de la couche MAC. L'attribut peut contenir jusqu'à deux clés de 16 octets. La valeur de l'identifiant de clé doit être différente pour chaque clé.

Pour des raisons de sécurité, les entrées de clé ne peuvent pas être lues, mais uniquement écrites.

```
array mac_GMK

mac_GMK::= structure

{
key_id:     unsigned,
key:   octet-string
}
```

Où:

key_id Identifiant de clé utilisé pour faire référence à cette clé, peut prendre la valeur 0 ou 1.

key Clé AES-128 utilisée pour le chiffrement des trames échangées au niveau de la couche MAC.

4.13.4.2.6 mac_frame_counter

Attribut PIB 0x0077: compteur de trames sortantes pour ce dispositif, utilisé lors du chiffrement des trames au niveau de la couche MAC.

4.13.4.2.7 mac_tone_mask

Attribut PIB 0x0110 Définit le masque de tonalité à utiliser lors de la formation du symbole.

4.13.4.2.8 mac_TMR_TTL

Attribut PIB 0x010D: durée de vie maximale de l'entrée des paramètres de mise en correspondance de tonalité dans la table de voisinage, en minutes.

NOTE Cet attribut a été utilisé avec différentes définitions dans les versions antérieures de la classe d'interfaces. Voir 5.13.6.2.8.

4.13.4.2.9 mac_max_frame_retries

Attribut PIB 0x0059: nombre maximal de retransmissions.

4.13.4.2.10 mac_neighbour_table_entry_TTL

Attribut PIB 0x010E: durée de vie maximale d'une entrée dans la table de voisinage, en minutes.

NOTE Cet attribut a été utilisé avec un nom différent et différentes définitions dans les versions antérieures de la classe d'interfaces. Voir 5.13.6.2.10.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2.11 mac_neighbour_table

Attribut PIB 0x010A: Voir UIT-T G.9903:2017, 9.3.7.2 pour les bandes CENELEC et FCC.

La table de voisinage contient des informations relatives à tous les dispositifs à l'intérieur du POS du dispositif. Un élément de la table représente un voisin direct PLC du dispositif.

```
array neighbour_table

neighbour_table::= structure

{
short_address:                  long-unsigned,
payload_modulation_scheme:      boolean,
tone_map:                       bit-string,
modulation_type:                    enum,
tx_gain:                        integer,
tx_res:                         enum,
tx_coeff:                       bit-string,
reverse_lqi:                        unsigned,
phase_differential              integer,
TMR_valid_time:                 unsigned,
no_data:                        unsigned
}
```

Où:

short_address — Adresse courte MAC du nœud auquel cette entrée fait référence.

payload_modulation_scheme — Schéma de modulation de charge utile à utiliser lors de la transmission vers ce voisin.
FALSE: différentiel,
TRUE: cohérent.

tone_map — Ce paramètre définit la sous-bande de fréquence qui peut être utilisée pour communiquer avec le dispositif. Une valeur 1 attribuée à un bit signifie que la sous-bande de fréquence peut être utilisée, une valeur 0 signifiant qu'elle ne doit pas l'être.

modulation — Type de modulation à utiliser pour communiquer avec le dispositif.

enum:

(0) Mode robuste,
(1) DBPSK,
(2) DQPSK,
(3) D8PSK,
(4) 16-QAM.

NOTE 1 La modulation 16-QAM est facultative et s'applique uniquement pour la bande FCC.

tx_gain — Définit le gain de l'émetteur à utiliser pour émettre des trames vers ce dispositif.

tx_res — Définit la résolution du gain de l'émetteur correspondant à un échelon de gain.
0: 6 dB, 1: 3 dB.

tx_coeff — Paramètre qui spécifie le gain de l'émetteur pour chaque groupe de tonalités représenté par un bit valide de mise en correspondance de tonalité.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

NOTE 2 Un groupe de tonalités rassemble 6 tonalités (ou porteuses) consécutives pour les bandes CENELEC, et 3 tonalités consécutives pour les bandes FCC.

Le récepteur mesure l'affaiblissement relatif à la fréquence du canal et peut demander à l'émetteur de le compenser en augmentant la puissance sur des parties du spectre confrontées à l'affaiblissement, afin d'égaliser le signal reçu. Chaque groupe de tonalités est mis en correspondance avec une valeur de 4 bits (CENELEC-A) ou de 2 bits (FCC). La valeur 0 du bit de poids fort indique une valeur de gain positive, une augmentation du gain de l'émetteur mis à l'échelle par TXRES étant donc demandée pour cette section. Une valeur 1 indique une valeur de gain négative, une diminution du gain de l'émetteur mis à l'échelle par TXRES étant donc demandée pour cette section. La mise en œuvre de cette fonction est facultative et est destinée aux canaux à sélection de fréquence. Si cette fonction n'est pas mise en œuvre, la valeur zéro doit être utilisée.

Dans la planification des bandes CENELEC, le premier bit de la chaîne binaire (bit 0) correspond à TXCOEF[0] et le dernier bit dans la chaîne binaire (bit 23) correspond à TXCOEF[23], les bits étant ordonnés en groupes tels que définis dans l'UIT-T G.9903 Amd. 1:2017, Tableau 9-21.

Dans la planification des bandes FCC, le premier bit de la chaîne binaire (bit 0) correspond à TXCOEF[0] et le dernier bit dans la chaîne binaire (bit 47) correspond à TXCOEF[47], les bits étant ordonnés en groupes tels que définis dans l'UIT-T G.9903 Amd. 1:2017, Tableau 9-22.

NOTE 3 Une description de tx_coeff est incluse dans l'UIT G.9903:2017, Tableaux 9-21 et 9-22.

Dans la planification des bandes CENELEC-A, chaque groupe de 4 bits du champ TXCOEF est associé à une sous-bande composée d'un groupe de 6 tonalités.

Numéro de bit	Champ TXCOEF	Sous-bande (kHz)
0	TXCOEF[0]	35,937 5 à 43,75
1	TXCOEF[1]	
2	TXCOEF [2]	
3	TXCOEF [3]	
4	TXCOEF [4]	45,312 5 à 53,125
5	TXCOEF [5]	
6	TXCOEF[6]	
7	TXCOEF[7]	
8	TXCOEF[8]	54,687 5 à 62,5
9	TXCOEF[9]	
10	TXCOEF[10]	
11	TXCOEF[11]	
12	TXCOEF[12]	64,062 5 à 71,875
13	TXCOEF[13]	
14	TXCOEF[14]	
15	TXCOEF[15]	
16	TXCOEF[16]	73,437 5 à 81,25

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Numéro de bit	Champ TXCOEF	Sous-bande (kHz)
17	TXCOEF[17]	
18	TXCOEF[18]	
19	TXCOEF[19]	
20	TXCOEF[20]	82,812 5 à 90,625
21	TXCOEF[21]	
22	TXCOEF[22]	
23	TXCOEF[23]	

Dans la planification des bandes FCC, chaque groupe de 2 bits du champ TXCOEF est associé à une sous-bande composée d'un groupe de 3 tonalités.

Numéro de bit	Champ TXCOEF	Sous-bande (kHz)
0	TXCOEF[0]	154,687 5 à 164,062 5
1	TXCOEF[1]	
2	TXCOEF [2]	168,75 à 178,125
3	TXCOEF [3]	
4	TXCOEF [4]	182,812 5 à 192,187 5
5	TXCOEF [5]	
6	TXCOEF[6]	196,875 à 206,25
7	TXCOEF[7]	
8	TXCOEF[8]	210,937 5 à 220,312 5
9	TXCOEF[9]	
10	TXCOEF[10]	225 à 234,375
11	TXCOEF[11]	
12	TXCOEF[12]	239,062 5 à 248,437 5
13	TXCOEF[13]	
14	TXCOEF[14]	253,125 à 262,5
15	TXCOEF[15]	
16	TXCOEF[16]	267,187 5 à 276,562 5
17	TXCOEF[17]	
18	TXCOEF[18]	281,25 à 290,625
19	TXCOEF[19]	
20	TXCOEF[20]	295,312 5 à 304,687 5
21	TXCOEF[21]	
22	TXCOEF[22]	309,375 à 318,75
23	TXCOEF[23]	
24	TXCOEF[24]	323,437 5 à 332,812 5
25	TXCOEF[25]	
26	TXCOEF[26]	337,5 à 346,875
27	TXCOEF[27]	
28	TXCOEF[28]	351,562 5 à 360,937 5
29	TXCOEF[29]	
30	TXCOEF[30]	365,625 à 375

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Numéro de bit	Champ TXCOEF	Sous-bande (kHz)
31	TXCOEF[31]	
32	TXCOEF[32]	379,687 5 à 389,062 5
33	TXCOEF[33]	
34	TXCOEF[34]	393,75 à 403,125
35	TXCOEF[35]	
36	TXCOEF[36]	407,812 5 à 417,187 5
37	TXCOEF[37]	
38	TXCOEF[38]	421,875 à 431,25
39	TXCOEF[39]	
40	TXCOEF[40]	435,937 5 à 445,312 5
41	TXCOEF[41]	
42	TXCOEF[42]	450 à 459,375
43	TXCOEF[43]	
44	TXCOEF[44]	464,062 5 à 473,437 5
45	TXCOEF[45]	
46	TXCOEF[46]	478,125 à 487,5
47	TXCOEF[47]	

reverse_lqi — Indicateur qualité de liaison de la liaison vers le voisin (LQI inverse)

NOTE 4 Valeur LQI mesurée lors de la réception de la PPDU provenant du voisin. La mesure de LQI caractérise la puissance et/ou la qualité d'un paquet reçu.

phase_differential — Déphasage en multiples de 60 degrés entre la phase du secteur du nœud local et le nœud voisin. PhaseDifferential peut supposer six valeurs entières comprises entre 0 et 5.

TMR_valid_time — Temps restant, en minutes, pendant lequel les paramètres de réponse de mise en correspondance de tonalité dans la table de voisinage sont considérés comme étant valides.

– Lors de la création de l'entrée, la valeur 0 doit être attribuée par défaut à cette valeur.
– A 0, une requête de mise en correspondance de tonalité peut être formulée si des données sont envoyées à ce dispositif. Dès la réception d'une réponse de mise en correspondance de tonalité, cette valeur est définie sur mac_TMR_TTL.

no_data — Valeur qui peut être utilisée selon les besoins pour certaines mises en œuvre déterminées par une spécification d'accompagnement

4.13.4.2.12 mac_high_priority_window_size

Attribut PIB 0x0100: taille de la fenêtre de contention à priorité élevée en nombre d'intervalles.

4.13.4.2.13 mac_CSMA_fairness_limit

Attribut PIB 0x010C: limite de fidélité d'accès au canal. Spécifie le nombre de tentatives de repli qui n'ont pas abouti, l'exposant de repli étant défini sur minBE. Une valeur comprise entre 2 × (macMaxBE – macMinBE) et 255 peut être affectée à cet attribut.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2.14 mac_beacon_randomization_window_length

Attribut PIB 0x0111: durée, en secondes, de randomisation de l'étiquette.

4.13.4.2.15 mac_A

Attribut PIB 0x0112: ce paramètre commande la diminution linéaire CW adaptative.

4.13.4.2.16 mac_K

Attribut PIB 0x0113: facteur d'adaptation au débit binaire pour la limite de fidélité d'accès au canal. Une valeur comprise entre 1 et macCSMAFairnessLimit peut être affectée à cet attribut.

4.13.4.2.17 mac_min_CW_attempts

Attribut PIB 0x0114: nombre de tentatives consécutives lors de l'utilisation de CW minimal.

4.13.4.2.18 mac_cenelec_legacy_mode

Attribut PIB 0x0115: cet attribut en lecture seule indique la capacité du nœud.

0: la configuration suivante est utilisée (mode existant):

- entrelaçage élémentaire;
- les paramètres d'entrelaceur n_i et n_j ne sont pas permutés lorsque I(i,j) = 0;

1: la configuration suivante est utilisée (mode non existant):

- entrelaçage de bloc complet;
- les paramètres d'entrelaceur n_i et n_j sont permutés lorsque I(i,j) = 0.

4.13.4.2.19 mac_FCC_legacy_mode

Attribut PIB 0x0116: cet attribut en lecture seule indique la capacité du nœud.

0: la configuration suivante est utilisée (mode existant):

- modulation FCH différentielle;
- entrelaçage élémentaire;
- les paramètres d'entrelaceur n_i et n_j ne sont pas permutés lorsque I(i,j) = 0;
- bloc RS simple;

1: la configuration suivante est utilisée (mode non existant):

- modulation FCH cohérente;
- entrelaçage de bloc complet;
- les paramètres d'entrelaceur n_i et n_j sont permutés lorsque I(i,j) = 0;
- deux blocs RS.

4.13.4.2.20 mac_max_BE

Attribut PIB 0x0047: valeur maximale de l'exposant de repli. Il convient qu'il soit toujours supérieur à macMinBE.

4.13.4.2.21 mac_max_CSMA_backoff

Attribut PIB 0x004E: nombre maximal de tentatives de repli.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.4.2.22 mac_min_BE

Attribut PIB 0x004F: valeur minimale de l'exposant de repli.

4.13.4.3 Description de la méthode

4.13.4.3.1 mac_get_neighbour_table_entry (data)

Cette méthode est utilisée pour récupérer la table de voisinage MAC correspondant à une adresse courte MAC. Elle peut être utilisée pour permettre au client de surveiller la topologie.

Le paramètre d'appel de méthode contient une mac_short_address.

data::= long-unsigned

Le paramètre de réponse inclut la table de voisinage pour cette mac_short_address.

data::= array neighbour_table

où mac_neighbour_table est tel que défini dans l'attribut *mac_neighbour_table* de la présente IC.

4.13.5 G3-PLC 6LoWPAN adaptation layer setup (class_id = 92, version = 2)

4.13.5.1 Vue d'ensemble

Une instance de l'IC "G3-PLC 6LoWPAN adaptation layer setup" contient les paramètres nécessaires à l'établissement et la gestion de la couche adaptation G3-PLC 6LoWPAN.

Ces attributs influent sur le comportement fonctionnel d'une mise en œuvre. Les mises en œuvre peuvent permettre de modifier leurs valeurs pendant le fonctionnement normal, c'est-à-dire même si la séquence de démarrage du dispositif a été exécutée.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

G3-PLC 6LoWPAN adaptation layer setup		0...n	class_id = 92, version = 2			
Attribut(s)		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1.	logical_name (static)	octet-string				x
2.	adp_max_hops (static)	unsigned	1	14	8	x + 0x08
3.	adp_weak_LQI_value (static)	unsigned	0	255	52	x + 0x10
4.	adp_security_level (static)	unsigned	0	5	5	x + 0x18
5.	adp_prefix_table (dyn)	array				x + 0x20
6.	adp_routing_configuration (static)	array				x + 0x28
7.	adp_broadcast_log_table_entry_TTL (static)	long-unsigned	0	65535	2	x + 0x30
8.	adp_routing_table (dyn)	array				x + 0x38
9.	adp_context_information_table (dyn)	array				x + 0x40
10.	adp_blacklist_table (dyn)	array				x + 0x48
11.	adp_broadcast_log_table (dyn)	array				x + 0x50
12.	adp_group_table (dyn)	array				x + 0x58
13.	adp_max_join_wait_time (static)	long-unsigned	0	1023	20	x + 0x60
14.	adp_path_discovery_time (static)	unsigned	0	255	40	x + 0x68
15.	adp_active_key_index (static)	unsigned	0	1	0	x + 0x70
16.	adp_metric_type (static)	unsigned	0x00	0x0F	0x0F	x + 0x78
17.	adp_coord_short_address (static)	long-unsigned	0x0000	0x7FFF	0x0000	x + 0x80
18.	adp_disable_default_routing (static)	boolean			FALSE	x + 0x88
19.	adp_device_type (static)	enum	0	2	2	x + 0x90
20.	adp_default_coord_route_enabled (static)	boolean			FALSE	x + 0x98
21.	adp_destination_address_set (dyn)	array				x + 0xA0
Méthodes spécifiques		***o/f***				

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.5.2 Description d'attribut

4.13.5.2.1 logical_name

Identifie l'instance de l'objet "G3-PLC OFDM 6LoWPAN adaptation layer setup". Voir 6.2.28.

4.13.5.2.2 adp_max_hops

Attribut PIB 0x0F: définit le nombre maximal de sauts à utiliser par l'algorithme d'acheminement.

4.13.5.2.3 adp_weak_LQI_value

Attribut PIB 0x1A: la valeur de liaison faible définit la valeur LQI en deçà de laquelle une liaison vers un voisin est considérée comme étant une liaison faible. Une valeur de 52 représente un SNR de 3 dB.

4.13.5.2.4 adp_security_level

Attribut PIB 0x00: niveau de sécurité minimal à utiliser pour les trames d'adaptation entrantes et sortantes. Seules les valeurs 0 (pas de chiffrement) et 5 (chiffrement avec un code d'intégrité de 32 bits) sont prises en charge.

4.13.5.2.5 adp_prefix_table

Attribut PIB 0x01: contient la liste des préfixes définis pour ce réseau PAN.

NOTE Il est pris pour hypothèse que l'adresse IPv6 locale de la liaison existe indépendamment et n'est pas affectée par les préfixes définis dans la table des préfixes.

4.13.5.2.6 adp_routing_configuration

L'élément de configuration d'acheminement spécifie tous les paramètres liés au mécanisme d'acheminement décrits dans l'UIT-T G.9903 Amd. 1:2017. Les éléments sont spécifiés au 9.4.1.2 de cette Recommandation.

NOTE 1 Le calcul du coût de liaison est présenté dans l'UIT-T G.9903:2017, Annexe B.

```
array routing_configuration
routing_configuration::= structure
{
adp_net_traversal_time:              unsigned,
adp_routing_table_entry_TTL:         long-unsigned,
adp_Kr:                              unsigned,
adp_Km:                              unsigned,
adp_Kc:                              unsigned,
adp_Kq:                              unsigned,
adp_Kh:                              unsigned,
adp_Krt:                             unsigned,
adp_RREQ_retries:                    unsigned,
adp_RREQ_wait:                       unsigned,
adp_blacklist_table_entry_TTL:       long-unsigned,
adp_unicast_RREQ_gen_enable:         boolean,
adp_RLC_Enabled:                     boolean,
adp_add_rev_link_cost:               unsigned
}
```

Où:

adp_net_traversal_time — Attribut PIB 0x11: durée maximale qu'un paquet est censé mettre pour atteindre un nœud à partir d'un autre nœud, en secondes.
Plage : 0-255
Valeur par défaut : 20

adp_routing_table_entry_TTL — Attribut PIB 0x12: durée de vie maximale d'une entrée de la table d'acheminement (en minutes).
Plage : 0-65 535
Valeur par défaut : 60

adp_Kr — Attribut PIB 0x13: facteur de pondération du mode robuste pour calculer le coût de liaison.
Plage : 0-31
Valeur par défaut : 0

adp_Km — Attribut PIB 0x14: facteur de pondération de modulation pour calculer le coût de liaison.
Plage : 0-31
Valeur par défaut : 0

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

adp_Kc — Attribut PIB 0x15: facteur de pondération du nombre de tonalités actives pour calculer le coût de liaison.
Plage : 0-31
Valeur par défaut : 0

adp_Kq — Attribut PIB 0x16: facteur de pondération de LQI pour calculer le coût d'acheminement.
Plage : 0-50
Valeur par défaut : 10 pour la bande CENELEC A / 40 pour la bande FCC.

adp_Kh — Attribut PIB 0x17: facteur de pondération du saut pour calculer le coût de liaison.
Plage : 0-31
Valeur par défaut : 4 pour la bande CENELEC A / 2 pour la bande FCC.

adp_Krt — Attribut PIB 0x1B: facteur de pondération du nombre de chemins actifs dans la table d'acheminement pour calculer le coût de liaison.
Plage : 0-31
Valeur par défaut : 0

adp_RREQ_retries — Attribut PIB 0x18: nombre de retransmissions RREQ en cas d'expiration du délai de réception RREP.
Plage : 0-255
Valeur par défaut : 0

adp_RREQ_wait — Attribut PIB 0x19: nombre de secondes d'attente entre deux générations consécutives de RREQ – RERR.
Plage : 0-255
Valeur par défaut : 30

NOTE 2 Le titre et la définition de cet élément ont été modifiés par rapport à la version précédente.

adp_blacklist_table_entry_TTL — Attribut PIB 0x1F: durée de vie maximale d'une entrée d'un ensemble de voisins sur liste noire (en minutes).
Plage : 0-65 535
Valeur par défaut : 10

adp_unicast_RREQ_gen_enable — Attribut PIB 0x0D: Si la valeur est TRUE, le RREQ doit être généré en attribuant la valeur 1 à son fanion "unicast RREQ".
Si la valeur est FALSE, le RREQ doit être généré en attribuant la valeur 0 à son fanion "unicast RREQ".
Valeur par défaut : TRUE

adp_RLC_enabled — Attribut PIB 0x09: active l'envoi de la trame RLCREQ par le dispositif.
Valeur par défaut : FALSE

adp_add_rev_ink_cost — Attribut PIB 0x0A: représente un coût supplémentaire pour prendre en compte une éventuelle asymétrie dans la liaison.
Plage : 0-255
Valeur par défaut : 0

4.13.5.2.7 adp_broadcast_log_table_entry_TTL

Attribut PIB 0x02: durée de vie maximale d'une entrée adpBroadcastLogTable (en minutes).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.5.2.8 adp_routing_table

Attribut PIB 0x0C: contient la table d'acheminement.

```
array routing_table

routing_table::= structure

{
    destination_address:    long-unsigned,
    next_hop_address:           long-unsigned,
    route_cost:             long-unsigned,
    hop_count:              unsigned,
    weak_link_count:        unsigned,
    valid_time:             long-unsigned
}
```

NOTE 1 Cette table est actualisée à chaque fois qu'un chemin est conçu ou mis à jour (déclenché par le trafic de données) et à chaque fois que le temporisateur TTL expire.

Où:

destination_address — Adresse de la destination.

next_hop_address — Adresse du prochain saut sur le chemin vers la destination.

route_cost — Coût de liaison cumulé sur le chemin vers la destination.

hop_count — Nombre de sauts du chemin sélectionné vers la destination.
Plage: 0-14

NOTE 2 Dans la pratique, la valeur admise maximale est limitée par adp_max_hops.

weak_link_count — Nombre de liaisons faibles vers la destination.
Plage: 0-14

NOTE 3 Dans la pratique, la valeur admise maximale est limitée par adp_max_hops.

valid_time — Temps restant, en minutes, jusqu'à ce que cette entrée de la table d'acheminement soit considérée comme étant valide.

4.13.5.2.9 adp_context_information_table

Attribut PIB 0x07: contient les informations de contexte associées à chaque champ d'extension CID. Voir UIT-T G.9903:2014. Les éléments sont spécifiés dans le Tableau 9-30 de cette Recommandation.

```
array context_information_table

context_information_table::= structure

{
    CID:                bit-string,
    context_length:     unsigned,
    context:            octet-string,
    C:                  boolean,
    valid_lifetime:     long-unsigned
}
```

Où:

CID — Correspond aux informations de contexte à 4 bits utilisées pour les adresses source et de destination (SCI, DCI).
Le premier bit (bit 0) de la chaîne binaire correspond au bit de poids faible du CID dans l'UIT-T G.9903:2014, Tableau 9-30.
Plage: 0x00-0x0F

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

context_length	Indique la longueur du contexte transporté (des contextes jusqu'à 128 bits peuvent être transportés). Plage: 0-128
context	Correspond au contexte transporté utilisé pour les besoins de la compression/décompression.
C	Indique si le contexte est valide pour être utilisé dans la compression. FALSE: Seule la décompression est admise, TRUE: La compression et la décompression sont admises Un contexte peut être utilisé uniquement pour les besoins de la décompression. De plus, il convient de se conformer aux recommandations formulées dans la RFC 6775 afin de tenir compte de la propagation du contexte vers tous les nœuds du réseau PAN.
valid_lifetime	Temps restant, en minutes, au cours duquel la table d'informations contextuelles est considérée comme étant valide. Il est mis à jour dès la réception du contexte conseillé. Plage: 0-65 535

4.13.5.2.10 adp_blacklist_table

Attribut PIB 0x1E: contient la liste des voisins placés sur la liste noire.

```
array blacklisted_neighbour_set

blacklisted_neighbour_set::= structure

{
    blacklisted_neighbour_address:    long-unsigned,
    valid_time:                       long-unsigned
}
```

Où:

blacklisted_neighbour_address	Adresse à 16 bits du voisin placé sur la liste noire.
valid_time	Temps restant, en minutes, jusqu'à ce que cette entrée dans la table des voisins sur liste noire soit considérée comme étant valide.

4.13.5.2.11 adp_broadcast_log_table

Attribut PIB 0x0B: contient la table des journaux de diffusion.

NOTE Cette table fournit une liste des paquets de diffusion récemment reçus par le dispositif.

```
array broadcast_log_table

broadcast_log_table::= structure

{
    source_address:       long-unsigned,
    sequence_number:            unsigned,
    valid_time:           long-unsigned
}
```

Où:

source_address	Adresse source à 16 bits d'un paquet de diffusion. Il s'agit de l'adresse de l'initiateur de la diffusion.
sequence_number	Numéro de séquence contenu dans l'en-tête BC0.
valid_time	Temps restant, en minutes, jusqu'à ce que cette entrée de la table des journaux de diffusion soit considérée comme étant valide.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.13.5.2.12 adp_group_table

Attribut PIB 0x0E: contient les adresses de groupe auxquelles le dispositif appartient.

array group_address

group_address::= long-unsigned

group_address Adresse de groupe à laquelle ce nœud a été abonné.

4.13.5.2.13 adp_max_join_wait_time

Attribut PIB 0x20: délai d'expiration d'accès au réseau, en secondes, pour LBD.

4.13.5.2.14 adp_path_discovery_time

Attribut PIB 0x21: délai d'expiration pour la reconnaissance de chemin d'accès, en secondes.

4.13.5.2.15 adp_active_key_index

Attribut PIB 0x22: index du GMK actif à utiliser pour la transmission de données.

4.13.5.2.16 adp_metric_type

Attribut PIB 0x03: type de métrique à utiliser pour les besoins de l'acheminement.

4.13.5.2.17 adp_coord_short_address

Attribut PIB 0x08: définit l'adresse courte du coordinateur.

4.13.5.2.18 adp_disable_default_routing

Attribut PIB 0xF0: si TRUE, l'acheminement par défaut (LOADng) est désactivé. Si FALSE, l'acheminement par défaut (LOADng) est activé.

4.13.5.2.19 adp_device_type

Attribut PIB 0x10: définit le type du dispositif connecté au modem:

enum:

(0) Dispositif PAN,
(1) Coordinateur PAN,
(2) Non défini.

4.13.5.2.20 adp_default_coord_route_enabled

Attribut PIB 0x24: si TRUE, la couche adaptation ajoute un acheminement par défaut au coordonnateur après l'achèvement réussi de la procédure d'amorçage. Si FALSE, aucun acheminement par défaut ne sera créé.

4.13.5.2.21 adp_destination_address_set

Attribut PIB 0x23: Contient la liste des adresses des dispositifs pour lesquels ce routeur LOADng permet la connectivité.

array D_address
D_address::= long-unsigned

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

Où:

D_Address Adresse à 16 bits d'un nœud de destination associé à ce routeur LOADng et pour lequel ce routeur LOADng permet la connectivité.

4.14 Classes d'interfaces pour l'établissement et la gestion des réseaux de voisinage DLMS®/COSEM HS-PLC selon l'ISO/IEC 12139-1

4.14.1 Vue d'ensemble

Les objets COSEM destinés à l'échange de données à l'aide des réseaux de voisinage DLMS®/COSEM HS-PLC selon l'ISO/IEC 12139-1, s'ils sont mis en place, doivent se trouver dans le dispositif logique de gestion des serveurs COSEM.

Pour établir et gérer les réseaux de voisinage DLMS®/COSEM HS-PLC selon l'ISO/IEC 12139-1, les IC suivantes sont spécifiées:

- HS-PLC ISO/IEC 12139-1 MAC setup (voir 4.14.2);
- HS-PLC ISO/IEC 12139-1 CPAS setup (voir 4.14.3);
- HS-PLC ISO/IEC 12139-1 IP SSAS setup (voir 4.14.4);
- HS-PLC ISO/IEC 12139-1 HDLC SSAS setup (voir 4.14.5).

4.14.2 HS-PLC ISO/IEC 12139-1 MAC setup (class_id = 140, version = 0)

4.14.2.1 Vue d'ensemble

Les instances de l'IC "HS-PLC ISO/IEC 12139-1 MAC setup" contiennent les paramètres nécessaires à l'établissement et la gestion de la couche MAC du profil HS-PLC ISO/IEC 12139-1.

HS-PLC ISO/IEC 12139-1 MAC setup		**0...n**	**class_id = 140 version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name		octet-string				x
2. group_id	(static)	long64-unsigned	0	2^{46}		x + 0x08
3. secondary_group_id	(static)	long64-unsigned	0	2^{46}		x + 0x10
4. station_id	(static)	long64-unsigned	0	2^{48}		x + 0x18
5. parent_ station_id	(static)	long64-unsigned	0	2^{48}		x + 0x20
6. repeater_status	(static)	boolean				x + 0x28
7. encryption_mode	(static)	enum	1	2	1	x + 0x30
8. initial_encryption_key	(static)	octet-string				x + 0x38
9. rts/cts	(static)	boolean	0	1	0	x + 0x40
Méthodes spécifiques		***o/f***				

4.14.2.2 Description d'attribut

4.14.2.2.1 logical_name

Identifie l'instance de l'objet "HS-PLC ISO/IEC 12139-1 MAC setup". Voir 6.2.31.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.14.2.2.2 group_id

Contient l'identificateur de groupe de HS-PLC ISO/IEC 12139-1. Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

4.14.2.2.3 secondary_group_id

Contient l'identificateur de groupe secondaire de HS-PLC ISO/IEC 12139-1. Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

4.14.2.2.4 station_id

Contient l'identificateur de station de HS-PLC ISO/IEC 12139-1. Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

4.14.2.2.5 parent_station_id

Contient l'identificateur de station parent* de HS-PLC ISO/IEC 12139-1. Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

* La station parent est la station directement voisine de la station elle-même dans une liaison HS-PLC ISO/IEC 12139-1 à plusieurs étages de la station jusqu'à NNAP (une station source – un ou plusieurs Répéteurs – NNAP).

4.14.2.2.6 repeater_status

Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

Contient l'état de répéteur en cours du dispositif.

boolean:

FALSE = pas de répéteur,
TRUE = répéteur

4.14.2.2.7 encryption_mode

Contient le mode de cryptage utilisé par la station PLC. Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

enum:

(0) AES128,
(1) Réservé.

4.14.2.2.8 initial_encryption_key

Clé de cryptage utilisée à l'origine pendant la période d'enregistrement PLC (accès à la cellule). Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023

4.14.2.2.9 rts/cts

Contient l'état du paramètre RTS/CTS. Le paramètre RTS/CTS (Request To Send/Clear To Send) permet d'éviter la collision provoquée par les stations HS-PLC ISO/IEC 12139-1 masquées). Voir ISO/IEC 12139-1:2009, Article 7, pour plus de détails.

RTS/CTS activé/désactivé

boolean:
FALSE = désactivé,
TRUE = activé.

4.14.3 HS-PLC ISO/IEC 12139-1 CPAS setup (class_id = 141, version = 0)

4.14.3.1 Vue d'ensemble

Les instances de l'IC "HS-PLC ISO/IEC 12139-1 CPAS setup" contiennent les paramètres nécessaires à l'établissement et la gestion de la couche CPAS du profil HS-PLC ISO/IEC 12139-1.

HS-PLC ISO/IEC 12139-1 CPAS setup		**0...n**	**class_id = 141 version = 0**			
Attributs		***Type de données***	***Min.***	***Max.***	***Def.***	**Nom court**
1. logical_name		octet-string				x
2. cpas_address	(static)	long64-unsigned	0	2^{48}		x + 0x08
3. cpas_ether_type	(static)	long-unsigned				x + 0x10
4. master_station_cpas_address	(static)	long64-unsigned	0	2^{48}		x + 0x18
Méthodes spécifiques		**o/f**				

4.14.3.2 Description d'attribut

4.14.3.2.1 logical_name

Identifie l'instance de l'objet "HS-PLC ISO/IEC 12139-1 CPAS setup". Voir 6.2.31.

4.14.3.2.2 cpas_address

Contient l'adresse CPAS du profil HS-PLC ISO/IEC 12139-1.

Voir IEC 62056-8-6:2017, 5.4.2.

4.14.3.2.3 cpas_ether_type

Contient la valeur EtherType de la sous-couche CPAS.

Voir IEC 62056-8-6:2017, 5.4.2.

4.14.3.2.4 master_station_cpas_address

Contient l'adresse CPAS de la station maîtresse* du profil HS-PLC ISO/IEC 12139-1.

* La station maîtresse est la station de destination de la trame CPAS (c'est-à-dire NNAP, la plupart du temps) dans une liaison HS-PLC ISO/IEC 12139-1 à plusieurs étages (une station source – un ou plusieurs Répéteurs – une station de destination).

IECNORM.COM : Click to view the full PDF of IEC 62056-6-2:2023