

Communication networks and systems for power utility automation

Basic information and communication structure —
Abstract communication service interface (ACSI)

National foreword

This British Standard is the UK implementation of EN 61850-7-2:2010+A1:2020. It is derived from IEC 61850-7-2:2010 incorporating amendment A1:2020. It supersedes BS EN 61850-7-2:2010, which is withdrawn.

The text of amendment 1:2021 has been provided in its entirety at the end of this document. BSI's policy of providing consolidated content remains unchanged; however, in the interest of expediency, in this instance BSI have chosen to collate the relevant content at the end of this document.

The UK participation in its preparation was entrusted to Technical Committee PEL/57, Power systems management and associated information exchange.

A list of organizations represented on this committee can be obtained on request to its committee manager.

Contractual and legal considerations

This publication has been prepared in good faith, however no representation, warranty, assurance or undertaking (express or implied) is or will be made, and no responsibility or liability is or will be accepted by BSI in relation to the adequacy, accuracy, completeness or reasonableness of this publication. All and any such responsibility and liability is expressly disclaimed to the full extent permitted by the law.

This publication is provided as is, and is to be used at the recipient's own risk.

The recipient is advised to consider seeking professional guidance with respect to its use of this publication.

This publication is not intended to constitute a contract. Users are responsible for its correct application.

© The British Standards Institution 2022
Published by BSI Standards Limited 2022

ISBN 978 0 539 21761 2

ICS 33.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 January 2011.

Amendments/corrigenda issued since publication

Date	Text affected
30 April 2020	Implementation of IEC amendment 1:2020 with CENELEC endorsement A1:2020
30 April 2022	Correction of typesetting error: IEC amendment 1:2020 with CENELEC endorsement A1:2020 collated at end of standard

**Communication networks and systems for power utility automation -
Part 7-2: Basic information and communication structure -
Abstract communication service interface (ACSI)
(IEC 61850-7-2:2010)**

Réseaux de communication pour
l'automatisation des systèmes des
compagnies d'électricité -
Partie 7-2: Principes des structures
d'informations et de communication -
Interface de services abstraits de
communication (ACSI)
(CEI 61850-7-2:2010)

Kommunikationsnetze und -systeme für
die Automatisierung in der elektrischen
Energieversorgung -
Teil 7-2: Grundlegende Informations- und
Kommunikationsstruktur -
Abstrakte Schnittstelle für
Kommunikationsdienste (ACSI)
(IEC 61850-7-2:2010)

This European Standard was approved by CENELEC on 2010-10-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 57/1065/FDIS, future edition 2 of IEC 61850-7-2, prepared by IEC TC 57, Power systems management and associated information exchange, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 61850-7-2 on 2010-10-01.

This European Standard supersedes EN 61850-7-2:2003.

The major technical changes with regard to EN 61850-7-2:2003 are as follows:

- class diagrams have been updated;
- data types not required have been removed;
- errors and typos have been corrected;
- substitution model has been moved to EN 61850-7-3;
- service tracking for control blocks have been added;
- the view concept will be according to the new work on role based access (RBA);
- security issues are solved by the IEC 62351 series; and
- several terms have been harmonized with those in the other parts.

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

In this document, the following print types are used:

- **bold** is used to highlight defined terms;
- Tahoma is used where the difference between a capital i (I) and a small L (l) is important to see.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN and CENELEC shall not be held responsible for identifying any or all such patent rights.

The following dates were fixed:

- | | | |
|--|-------|------------|
| – latest date by which the EN has to be implemented at national level by publication of an identical national standard or by endorsement | (dop) | 2011-07-01 |
| – latest date by which the national standards conflicting with the EN have to be withdrawn | (dow) | 2013-10-01 |

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 61850-7-2:2010 was approved by CENELEC as a European Standard without any modification.

In the official version, for Bibliography, the following notes have to be added for the standards indicated:

- | | | |
|--------------------|------|---|
| IEC 61400-25-2 | NOTE | Harmonized as EN 61400-25-2. |
| IEC 61850-8 series | NOTE | Harmonized in EN 61850-8 series (not modified). |
| IEC 61850-9 series | NOTE | Harmonized in EN 61850-9 series (not modified). |

IEC 61850-9-1

NOTE Harmonized as EN 61850-9-1.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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

NOTE When an international publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC/TS 61850-2	-	Communication networks and systems in substations - Part 2: Glossary	-	-
IEC 61850-5	-	Communication networks and systems in substations - Part 5: Communication requirements for functions and device models	EN 61850-5	-
IEC 61850-6	-	Communication networks and systems for power utility automation - Part 6: Configuration description language for communication in electrical substations related to IEDs	EN 61850-6	-
IEC 61850-7-1	-	Communication networks and systems in substations - Part 7-1: Basic communication structure for substation and feeder equipment - Principles and models	EN 61850-7-1	-
IEC 61850-7-3	-	Communication networks and systems in substations - Part 7-3: Basic communication structure for substation and feeder equipment - Common data classes	EN 61850-7-3	-
IEC 61850-7-4	-	Communication networks and systems for power utility automation - Part 7-4: Basic communication structure - Compatible logical node classes and data object classes	EN 61850-7-4	-
IEC 61850-8-1	-	Communication networks and systems in substations - Part 8-1: Specific Communication Service Mapping (SCSM) - Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3	EN 61850-8-1	-
IEC 61850-9-2	-	Communication networks and systems in substations - Part 9-2: Specific Communication Service Mapping (SCSM) - Sampled values over ISO/IEC 8802-3	EN 61850-9-2	-

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
ISO 4217	-	Codes for the representation of currencies and funds	-	-
ISO 9506-1	-	Industrial automation systems - Manufacturing Message Specification - Part 1: Service definition	-	-
IEEE 754	-	Binary floating-point arithmetic	-	-

CONTENTS

FOREWORD.....	9
INTRODUCTION.....	11
1 Scope.....	12
2 Normative references.....	12
3 Terms and definitions	13
4 Abbreviated terms.....	14
5 ACSI overview and basic concepts.....	15
5.1 Conceptual model of IEC 61850.....	15
5.2 The meta-meta model.....	16
5.3 The meta model.....	16
5.3.1 General	16
5.3.2 Information modelling classes	17
5.3.3 Information exchange modelling classes	18
5.3.4 Relations between classes	20
5.4 The domain type model.....	21
5.5 The data instance model.....	21
6 TypeDefinitions.....	22
6.1 General	22
6.1.1 BasicTypes.....	22
6.1.2 CommonACSI Types.....	23
7 GenServerClass model	29
7.1 GenServerClass definition	29
7.1.1 GenServerClass syntax.....	29
7.1.2 GenServerClass attributes	30
7.2 Server class services.....	30
7.2.1 Overview of directory and GetDefinition services	30
7.2.2 GetServerDirectory	31
8 Application association model	32
8.1 Introduction	32
8.2 Concept of application associations	32
8.3 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model.....	32
8.3.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition	32
8.3.2 Two-party application association services	34
8.4 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class	37
8.4.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition.....	37
8.4.2 MULTICAST-Application-association (MCAA) class attributes	37
9 GenLogicalDeviceClass model.....	38
9.1 GenLogicalDeviceClass definition	38
9.1.1 GenLogicalDeviceClass syntax	38
9.1.2 GenLogicalDeviceClass attributes	38
9.2 GenLogicalDeviceClass services.....	38
9.2.1 GetLogicalDeviceDirectory	38
10 GenLogicalNodeClass model	39
10.1 GenLogicalNodeClass definition.....	39
10.1.1 GenLogicalNodeClass diagram	39
10.1.2 GenLogicalNodeClass syntax.....	41

10.1.3	GenLogicalNodeClass attributes	41
10.2	GenLogicalNodeClass services	42
10.2.1	Overview	42
10.2.2	GetLogicalNodeDirectory	42
10.2.3	GetAllDataValues	43
11	Generic data object class model	45
11.1	GenDataObjectClass diagram	45
11.2	GenDataObjectClass syntax	45
11.3	GenDataObjectClass attributes	46
11.3.1	DataObjectName	46
11.3.2	DataObjectRef – data object reference	46
11.3.3	m/o/c	46
11.3.4	DataObjectType	46
11.4	GenDataObjectClass services	46
11.4.1	General definitions and overview	46
11.4.2	GetDataValues	47
11.4.3	SetDataValues	48
11.4.4	GetDataDirectory	49
11.4.5	GetDataDefinition	50
12	Generic common data class model	50
12.1	General	50
12.2	GenCommonDataClass	51
12.2.1	GenCommonDataClass diagram	51
12.2.2	GenCommonDataClass syntax	51
12.2.3	GenCommonDataClass attributes	52
12.3	GenDataAttributeClass	52
12.3.1	GenDataAttributeClass diagram	52
12.3.2	GenDataAttributeClass syntax	53
12.3.3	GenDataAttributeClass attributes	53
12.4	GenConstructedAttributeClass	57
12.4.1	GenConstructedAttributeClass diagram	57
12.4.2	GenConstructedAttributeClass syntax	57
12.4.3	GenConstructedAttributeClass attributes	57
12.5	GenSubDataAttributeClass	57
12.5.1	SubDataAttributeClass diagram	57
12.5.2	SubDataAttributeClass syntax	58
12.5.3	GenSubDataAttributeClass attributes	58
12.6	Referencing data objects and their components	58
12.6.1	General	58
12.6.2	Reference syntax	59
12.6.3	Base types and their relation	59
12.6.4	Example of using references	60
13	DATA-SET class model	61
13.1	General	61
13.2	DATA-SET class definition	62
13.2.1	DATA-SET class syntax	62
13.2.2	DATA-SET class attributes	63
13.3	DATA-SET class services	63
13.3.1	Overview	

13.3.2	GetDataSetValues	64
13.3.3	SetDataSetValues	65
13.3.4	CreateDataSet	66
13.3.5	DeleteDataSet	66
13.3.6	GetDataSetDirectory	67
14	Service tracking	68
14.1	General	68
14.2	Common service tracking (CST)	68
15	Modelling of control block classes	70
15.1	General	70
15.2	Control block class models	70
15.2.1	Control block attributes	71
15.2.2	Control block services	71
15.2.3	Attribute type	71
15.3	Control block tracking services	71
15.3.1	General	71
15.3.2	Common data classes for control block service tracking	72
16	SETTING-GROUP-CONTROL-BLOCK class model	82
16.1	General	82
16.2	SGCB class definition	83
16.2.1	SGCB class syntax	83
16.2.2	SGCB class attributes	84
16.3	SGCB class services	85
16.3.1	Overview	85
16.3.2	SelectActiveSG	85
16.3.3	SelectEditSG	86
16.3.4	SetEditSGValue	87
16.3.5	ConfirmEditSGValues	88
16.3.6	GetEditSGValue	89
16.3.7	GetSGCBValues	90
17	REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK class models	91
17.1	Overview	91
17.2	REPORT-CONTROL-BLOCK class model	93
17.2.1	Basic concepts	93
17.2.2	BUFFERED-REPORT-CONTROL-BLOCK (BRCB) class definition	93
17.2.3	BRCB class services	103
17.2.4	UNBUFFERED-REPORT-CONTROL-BLOCK (URCB) class definition	116
17.2.5	URCB class services	117
17.3	LOG-CONTROL-BLOCK class model	118
17.3.1	General	118
17.3.2	LCB class definition	119
17.3.3	LOG class definition	124
17.3.4	Reason code for log entries	127
17.3.5	LOG services	127
18	Generic substation event class model (GSE)	131
18.1	Overview	131
18.2	GOOSE-CONTROL-BLOCK (GoCB) class	132
18.2.1	GoCB definition	132
18.2.2	GOOSE service definitions	134

18.2.3	Generic object oriented substation event (GOOSE) message	139
19	Transmission of sampled value class model	140
19.1	Overview	140
19.2	Transmission of sampled values using multicast	142
19.2.1	MSVCB class definition	142
19.2.2	Multicast sampled value class services	144
19.3	Transmission of sampled values using unicast	147
19.3.1	USVCB class definition	147
19.3.2	Unicast sampled value services	150
19.4	Sampled value format	153
19.4.1	MsvID or UsvID	154
19.4.2	OptFlds	154
19.4.3	DatSet	154
19.4.4	Sample [1..n]	155
19.4.5	SmpCnt	155
19.4.6	RefrTm	155
19.4.7	ConfRev	155
19.4.8	SmpSynch	155
19.4.9	SmpRate	155
19.4.10	SmpMod	155
19.4.11	Simulation	155
20	CONTROL class model	156
20.1	Introduction	156
20.2	Control with normal security	158
20.2.1	Direct control with normal security	158
20.2.2	SBO control with normal security	160
20.3	Control with enhanced security	162
20.3.1	Introduction	162
20.3.2	Direct control with enhanced security	162
20.3.3	SBO control with enhanced security	163
20.4	Time-activated operate	166
20.5	CONTROL class service definitions	167
20.5.1	Overview	167
20.5.2	Service parameter definition	168
20.5.3	Service specification	172
20.6	Tracking of control services	178
20.6.1	General	178
20.6.2	Control service tracking (CTS)	178
21	Time and time-synchronization model	179
21.1	General	179
21.2	External information	180
22	Naming conventions	181
22.1	Class naming and class specializations	181
22.2	Referencing an instance of a class	182
22.3	Scope	183
23	File transfer model	184
23.1	File class	184
23.1.1	FileName	184
23.1.2	FileSize	184

23.1.3 LastModified	184
23.2 File services	185
23.2.1 GetFile	185
23.2.2 SetFile	185
23.2.3 DeleteFile	186
23.2.4 GetFileAttributeValues	186
Annex A (normative) ACSI conformance statement.....	188
Annex B (normative) Formal definition of IEC 61850-7-2 Common Data Classes.....	195
Annex C (informative) Generic substation state event (GSSE) control block (GsCB)	203
Bibliography	212
Index	213
Figure 1 – Excerpt of conceptual model of IEC 61850.....	16
Figure 2 – Basic conceptual class model of the ACSI.....	17
Figure 3 – Conceptual service model of the ACSI	19
Figure 4 – Core of the conceptual meta model and relationship	21
Figure 5 – Data instance model (conceptual).....	22
Figure 6 – Overview about GetDirectory and GetDefinition services	30
Figure 7 – Normal operation	33
Figure 8 – Aborting association	33
Figure 9 – Principle of multicast application association.....	37
Figure 10 – Basic conceptual model of the GenLogicalNodeClass.....	40
Figure 11 – Basic conceptual class model of the GenDataObjectClass	45
Figure 12 – Excerpt of GenDataObjectClass services	47
Figure 13 – Class diagram of the GenCommonDataClass	51
Figure 14 – Conceptual Class diagram of the GenCommonDataClass.....	51
Figure 15 – Class diagram of the GenDataAttributeClass.....	52
Figure 16 – Relation of TrgOp and Reporting.....	56
Figure 17 – Class diagram of the GenConstructedAttributeClass	57
Figure 18 – Relation of types (example)	60
Figure 19 – Example of a data object	61
Figure 20 – Dynamic creation of data set instances	62
Figure 21 – Control block service mapping	72
Figure 22 – Basic model of the settings model.....	83
Figure 23 – Basic building blocks for reporting and logging	92
Figure 24 – BRCB state machine.....	95
Figure 25 – General queue of entries for report handler	96
Figure 26 – Buffer time.....	98
Figure 27 – State Machine for Sequence Number Generation	99
Figure 28 – Logical state machine for general interrogation	101
Figure 29 – Report example on the use of sequence number	105
Figure 30 – Entry discard that does not cause indication of loss of information in enabled state	106
Figure 31 – Indication of loss of information due to resource constraints in enabled state	107

Figure 32 – Data set members and reporting	108
Figure 33 – Report example	109
Figure 34 – Log model overview	119
Figure 35 – GoCB model	131
Figure 36 – Model for transmission of sampled values	141
Figure 37 – Principle of the control model	156
Figure 38 – State machine of direct control with normal security	159
Figure 39 – Direct control with normal security	160
Figure 40 – State machine of SBO control with normal security	161
Figure 41 – State machine of direct control with enhanced security	163
Figure 42 – State machine SBO control with enhanced security	164
Figure 43 – Select before operate with enhanced security – positive case	165
Figure 44 – Select before operate with enhanced security – negative case (no status change)	165
Figure 45 – Time-activated operate	167
Figure 46 – Time model and time synchronization (principle)	180
Figure 47 – Specializations	181
Figure 48 – Object names and object reference	183
Figure C.1 – GsCB model	203
Table 1 – ACSI model classes with related services	20
Table 2 – BasicTypes	23
Table 3 – ObjectName type	24
Table 4 – ObjectReference type	24
Table 5 – ServiceError type	25
Table 6 – PACKED-LIST type	26
Table 7 – TimeStamp type	26
Table 8 – TimeQuality definition	27
Table 9 – TimeAccuracy	28
Table 10 – TriggerConditions type	28
Table 11 – ReasonForInclusion	29
Table 12 – GenServerClass definition	29
Table 13 – TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition	33
Table 14 – MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition	37
Table 15 – GenLogicalDeviceClass (GenLD) class definition	38
Table 16 – GenLogicalNodeClass definition	40
Table 17 – GenDataObjectClass definition	46
Table 18 – GenCommonDataClass definition	52
Table 19 – GenDataAttributeClass definition	53
Table 20 – Functional constraint values	54
Table 21 – TrgOp	56
Table 22 – GenConstructedAttributeClass definition	57
Table 23 – GenSubDataAttributeClass definition	58

Table 24 – DATA-SET (DS) class definition	63
Table 25 – Common service tracking common data class (CST) definition	69
Table 26 – ServiceType type	70
Table 27 – CB class definition	71
Table 28 – Buffered report tracking service (BTS) definition.....	73
Table 29 – Unbuffered report tracking service (UTS) definition	74
Table 30 – Log control block tracking service (LTS) definition.....	76
Table 31 – Log tracking service (OTS) definition.....	77
Table 32 – GOOSE Control block tracking service (GTS) definition.....	78
Table 33 – MSVCB tracking service (MTS) definition	79
Table 34 – USVCB tracking service (NTS) definition	80
Table 35 – SGCB tracking service (STS) definition	81
Table 36 – SGCB class definition	84
Table 37 – BRCB class definition	94
Table 38 – Report format specification	104
Table 39 – URCB class definition	116
Table 40 – LCB class definition	120
Table 41 – LOG class definition.....	125
Table 42 – GOOSE control block class definition	132
Table 43 – GOOSE message definition.....	139
Table 44 – MSVCB class definition	142
Table 45 – USVCB class definition	148
Table 46 – Sampled value (SV) format definition	154
Table 47 – Generic behavior and negative responses	157
Table 48 – Control services	167
Table 49 – T definition.....	168
Table 50 – Test definition	169
Table 51 – Check condition definition	169
Table 52 – operTm definition	169
Table 53 – Additional cause diagnosis definition	170
Table 54 – AddCause semantic	171
Table 55 – Control service tracking (CTS) definition	179
Table 56 – FILE class definition.....	184
Table A.1 – Basic conformance statement.....	189
Table A.2 – ACSI models conformance statement	190
Table A.3 – ACSI service conformance statement	191
Table C.1 – GSSE control block class definition	204
Table C.2 – GSSE message definition	210

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**COMMUNICATION NETWORKS AND SYSTEMS
FOR POWER UTILITY AUTOMATION –****Part 7-2: Basic information and communication structure –
Abstract communication service interface (ACSI)**

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.

International Standard IEC 61850-7-2 has been prepared by IEC technical committee 57: Power systems management and associated information exchange.

This bilingual version (2019-02) corresponds to the monolingual English version, published in 2010-08.

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

FDIS	Report on voting
57/1065/FDIS	57/1083/RVD

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

The French version of this standard has not been voted upon.

This second edition cancels and replaces the first edition published in 2003. It constitutes a technical revision.

Future standards in this series will carry the new general title as cited above. Titles of existing standards in this series will be updated at the time of the next edition.

The major technical changes with regard to the previous edition are as follows:

- class diagrams have been updated,
- data types not required have been removed,
- errors and typos have been corrected,
- substitution model has been moved to IEC 61850-7-3,
- service tracking for control blocks have been added,
- the view concept will be according to the new work on role bases access (RBA),
- security issues are solved by the IEC 62351 series, and
- several terms have been harmonized with those in the other parts.

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

In this document, the following print types are used:

- **bold** is used to highlight defined terms,
- Tahoma is used where the difference between a capital i (I) and a small L (l) is important to see.

A list of all parts of the IEC 61850 series, under the general title: *Communication networks and systems for power utility automation*, can be found on the IEC website.

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

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

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

INTRODUCTION

This document is part of a set of definitions which details a layered utility communication architecture. This architecture has been chosen to provide abstract definitions of classes and services such that the definitions are independent of specific protocol stacks, implementations, and operating systems.

The IEC 61850 series is intended to provide interoperability between a variety of devices. Communication between these devices is achieved by the definition of a hierarchical class model (for example, logical device, logical node, data, data set, report control, or log) and services provided by these classes (for example, get, set, report, define, delete) in IEC 61850-7-x.

This part of IEC 61850 defines the abstract communication service interface (ACSI) for use in the utility application domain that requires real-time cooperation of intelligent electronic devices. The ACSI has been defined so as to be independent of the underlying communication systems. Specific communication service mappings¹⁾ (SCSM) are specified in IEC 61850-8-x and IEC 61850-9-x.

This part of IEC 61850 defines the abstract communication service interface in terms of

- a hierarchical class model of all information that can be accessed via a communication network,
- services that operate on these classes, and
- parameters associated with each service.

The ACSI description technique abstracts away from all the different approaches to implement the cooperation of the various devices.

NOTE 1 Abstraction in ACSI has two meanings. First, only those aspects of a real device (for example, a breaker) or a real function that are visible and accessible over a communication network are modelled. This abstraction leads to the hierarchical class models and their behaviour defined in IEC 61850-7-2, IEC 61850-7-3, and IEC 61850-7-4. Second, the ACSI abstracts from the aspect of concrete definitions on how the devices exchange information; only a conceptual cooperation is defined. The concrete information exchange is defined in the SCSMs.

NOTE 2 This part of IEC 61850 does not provide comprehensive tutorial material. It is recommended that IEC 61850-5 and IEC 61850-7-1 be read first in conjunction with IEC 61850-7-2 and IEC 61850-7-3.

NOTE 3 Examples use names of classes (for example XCBR for a class of a logical node) defined in IEC 61850-7-4 and IEC 61850-7-3. The normative names are defined in IEC 61850-7-4 and IEC 61850-7-3 only.

1) The ACSI is independent of the specific mapping. Mappings to standard application layers or middle ware technologies are possible.

COMMUNICATION NETWORKS AND SYSTEMS FOR POWER UTILITY AUTOMATION –

Part 7-2: Basic information and communication structure – Abstract communication service interface (ACSI)

1 Scope

This part of IEC 61850 applies to the ACSI communication for utility automation. The ACSI provides the following abstract communication service interfaces.

- a) Abstract interface describing communications between a client and a remote server for
 - real-time data access and retrieval,
 - device control,
 - event reporting and logging,
 - setting group control,
 - self-description of devices (device data dictionary),
 - data typing and discovery of data types, and
 - file transfer.
- b) Abstract interface for fast and reliable system-wide event distribution between an application in one device and many remote applications in different devices (publisher/subscriber) and for transmission of sampled measured values (publisher/subscriber).

2 Normative references

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

IEC 61850-2, *Communication networks and systems in substations – Part 2: Glossary*

IEC 61850-5, *Communication networks and systems in substations – Part 5: Communication requirements for functions and devices models*

IEC 61850-6, *Communication networks and systems for power utility automation – Part 5: Configuration description language for communication in electrical substations related to IEDs*

IEC 61850-7-1, *Communication networks and systems for power utility automation – Part 7-1: Basic communication structure – Principles and models²⁾*

IEC 61850-7-3, *Communication networks and systems for power utility automation – Part 7-3: Basic communication structure – Common data classes²⁾*

IEC 61850-7-4, *Communication networks and systems for power utility automation – Part 7-4: Basic communication structure – Compatible logical node classes and data object classes*

²⁾ To be published.

IEC 61850-8-1, *Communication networks and systems for power utility automation – Part 8-1: Specific communication service mapping (SCSM) – Mappings to MMS (ISO 9506-1 and ISO 9506-2) and to ISO/IEC 8802-3³⁾*

IEC 61850-9-2, *Communication networks and systems for power utility automation – Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3³⁾*

ISO 4217, *Codes for the representation of currencies and funds*

ISO 9506 (all parts), *Industrial automation systems – Manufacturing Message Specification*

IEEE 754, *Standard for Floating-Point Arithmetic*

3 Terms and definitions

For the purposes of this document, the terms and definitions provided in IEC 61850-2 and the following apply.

3.1

class

description of a set of objects that share the same attributes, services, relationships, and semantics

3.2

client

entity that requests a service from a server and that receives unsolicited messages from a server

3.3

device

entity that performs control, actuating and/or sensing functions and interfaces to other such entities within an automation system

NOTE Devices alone do not perform energy generation, transport, or distribution functions.

3.4

external equipment

entity that is stand-alone, or interfaces to an automation system, and that performs energy generation, transport, or distribution functions

EXAMPLE Transformer, circuit-breaker, line.

NOTE 1 Equipment can contain devices.

NOTE 2 Equipment cannot have a direct connection to the communication network – only devices can be directly connected to the communication network.

3.5

instance (of a class)

entity that has unique identity, to which a set of services can be applied, and which has a state that stores the effects of the services

NOTE Instance is a synonym for the term object.

³⁾ To be published.

3.6

logical device

entity that represents a set of typical automation, protection or other functions

3.7

logical node

entity that represents a typical automation, protection or other function

3.8

physical device

entity that represents the physical parts of a device (hardware and operating system, etc.)

NOTE Physical devices host logical devices.

4 Abbreviated terms

AA	application association
ACSI	abstract communication service interface
BRCB	buffered report control block
CB	control block
CDC	common data class (IEC 61850-7-3)
CT	current transformer
DA	data attribute
DataRef	data reference
dchg	data change trigger option
DS	data set
dupd	data-update trigger option
FC	functional constraint
FCD	functional constrained data
FCDA	functional constrained data attribute
GI	general interrogation
GoCB	GOOSE control block
GOOSE	generic object oriented substation events
GSE	generic substation event
GsCB	GSSE control block
GSSE	generic substation status event
IED	intelligent electronic device
IntgPd	integrity period
LCB	log control block
LD	logical device (in this part of IEC 61850, the generic logical device class is defined (genLogicalDevice))
LN	logical node (in this part of IEC 61850, the generic logical node class is defined (genLogicalNode))
MC	multicast
MCAA	multicast application association
MMS	manufacturing message specification (ISO 9506)

MSVCB	multicast sampled value control block
PDU	protocol data unit
PICS	protocol implementation conformance statement
PIXIT	protocol Implementation extra information
qchg	quality change trigger option
SBO	select before operate
SCL	substation configuration language (IEC 61850-6)
SCSM	specific communication service mapping (defined in IEC 61850-8-x and IEC 61850-9-x)
SGCB	setting group control block
SoE	sequence-of-events
SVC	sampled value control
TP	two party
TPAA	two party application association
TrgOp	trigger option
UCA™ ⁴⁾	utility communication architecture
URCB	unbuffered report control block
UTC	coordinated universal time
SV	sampled value
USVCB	unicast sampled value control block
VT	voltage transformer

5 ACSI overview and basic concepts

5.1 Conceptual model of IEC 61850

The models of the ACSI provide

- the definition of the basic model of the utility information models contained in IEC 61850-7-3 (common data classes for utility automation applications), IEC 61850-7-4 (compatible logical node classes and compatible data classes for utility automation applications) and IEC 61850-6 (substation configuration language), and
- the definition of information exchange service models.

The information models and information exchange services are interwoven. From a descriptive point of view, the two aspects are separated to some degree (see the excerpt shown in Figure 1).

The first level of the definitions is a list of base types and rules how to build hierarchical structures (meta-meta model) defined in 5.2.

4) UCA™ is a registered trade mark of EPRI, Palo Alto (USA). This information is given for the convenience of users and does not constitute an endorsement by the IEC of the product named.

The generic models (meta model) for example, for logical nodes and data classes including their services, are defined 5.3 and applied in IEC 61850-7-3 and IEC 61850-7-4 to define many specialized information models for utility automation models or in IEC 61400-25-2 to define specialized models for wind power plant applications (domain type models; see 5.4).

The part IEC 61850-6 (SCL) defines the instances to be implemented (configured) in real devices (data instance model; see 5.5).

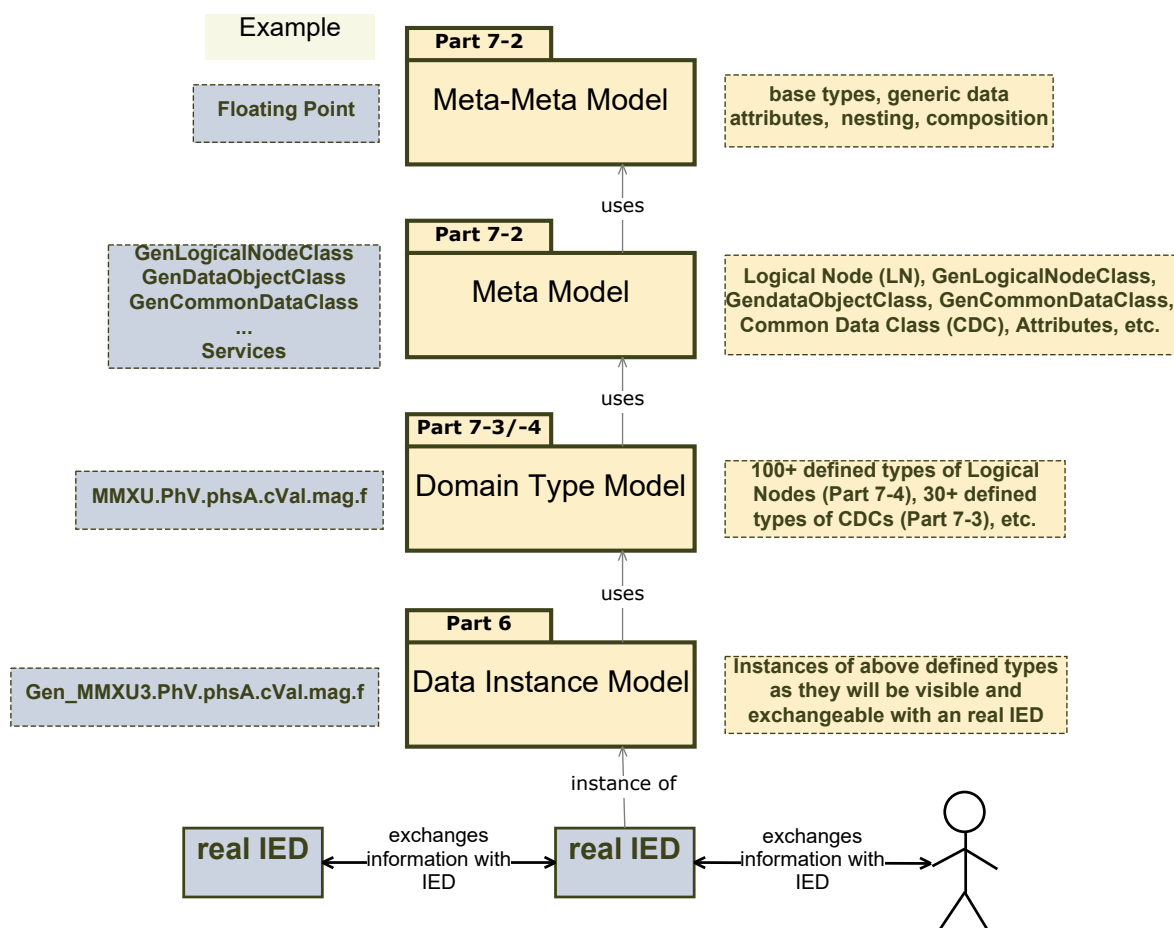


Figure 1 – Excerpt of conceptual model of IEC 61850

5.2 The meta-meta model

The meta-meta model is defined in Clause 6. It defines the basic data type classes to be used in the meta model with the exception of the recursion (nesting) of components to define hierarchical data models. The recursion is defined in the meta model.

5.3 The meta model

5.3.1 General

This part of IEC 61850 defines mainly the meta model for the whole IEC 61850 standard series. The meta model comprises classes for the description of a device with regard to data models and information exchange.

5.3.2 Information modelling classes

The following overall classes are defined:

- a) **Server** – represents the external visible behaviour of a device. All other ACSI models are part of the server.

NOTE A server has two roles: to communicate with a client (most service models in IEC 61850 provide communication with client devices) and to send information to peer devices (for example, for sampled values).

- b) **Logical device (LD)** – represents the information produced and consumed by a group of domain-specific application functions.
- c) **Logical node (LN)** – contains the information produced and consumed by a single domain-specific application function, for example, overvoltage protection or circuit-breaker.
- d) **Data objects** – provide means to define typed information, for example, position of a switch with quality information and timestamp, contained in logical nodes.

Each of these models is defined as a class. The classes comprise attributes and services. The conceptual class diagram of the ACSI is depicted in Figure 2.

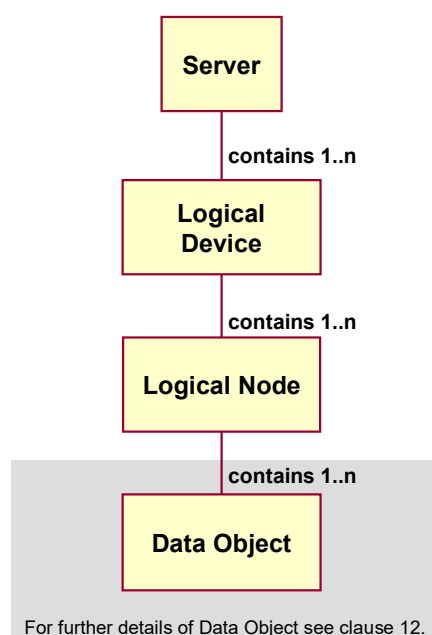


Figure 2 – Basic conceptual class model of the ACSI

Each of the following classes has a name and a reference: logical device, logical node, and data object.

EXAMPLE In an implementation, the logical device, logical node, data objects, and data attribute have each an object name (instance name) which is a unique name among classes of the same container to which they belong. In addition, each of the four has an ObjectReference (path name) which is a concatenation of all object names from each container. The four object names (one per column) can be concatenated.

	Logical device	Logical node	Data object	Data attribute
Object name	"Atlanta_HV5"	"XCBR1"	"Pos"	"stVal"
Description	High-voltage station 5	Circuit-breaker 1	Position	Status value

5.3.3 Information exchange modelling classes

In addition to the models listed above, the ACSI comprises the following models that provide services operating on data objects, data attributes, and data sets.

- a) **Data Set** – permits the grouping of data objects and data attributes. Used for direct access, for reporting, logging, GOOSE messaging and sampled value exchange.
- b) **Substitution** – supports replacement of a process value by another value.
- c) **Setting group control** – defines how to switch from one set of setting values to another one and how to edit setting groups.
- d) **Report control and logging** – describe the conditions for generating reports and logs based on parameters set by configuration or by a client. Reports may be triggered by changes of process data values (for example, state change or dead band) or by quality changes. Logs can be queried for later retrieval. Reports may be sent immediately or deferred. Reports provide change-of-state and sequence-of-events information exchange.
- e) **Control blocks for generic substation event (GSE)** – supports a fast and reliable system-wide distribution of input and output data values; peer-to-peer exchange of IED binary status information, for example, a trip signal.
- f) **Control blocks for transmission of sampled values** – fast and cyclic transfer of samples, for example, of instrument transformers.
- g) **Control** – describes the services to control, for example, devices.
- h) **Time and time synchronization** – provides the time base for the device and system.
- i) **File system** – defines the exchange of large data blocks such as programs.
- j) **Tracking** – provides a diagnosis interface to track services (control, configuration, exchange).

An overview of the conceptual service model of the ACSI is shown in Figure 3.

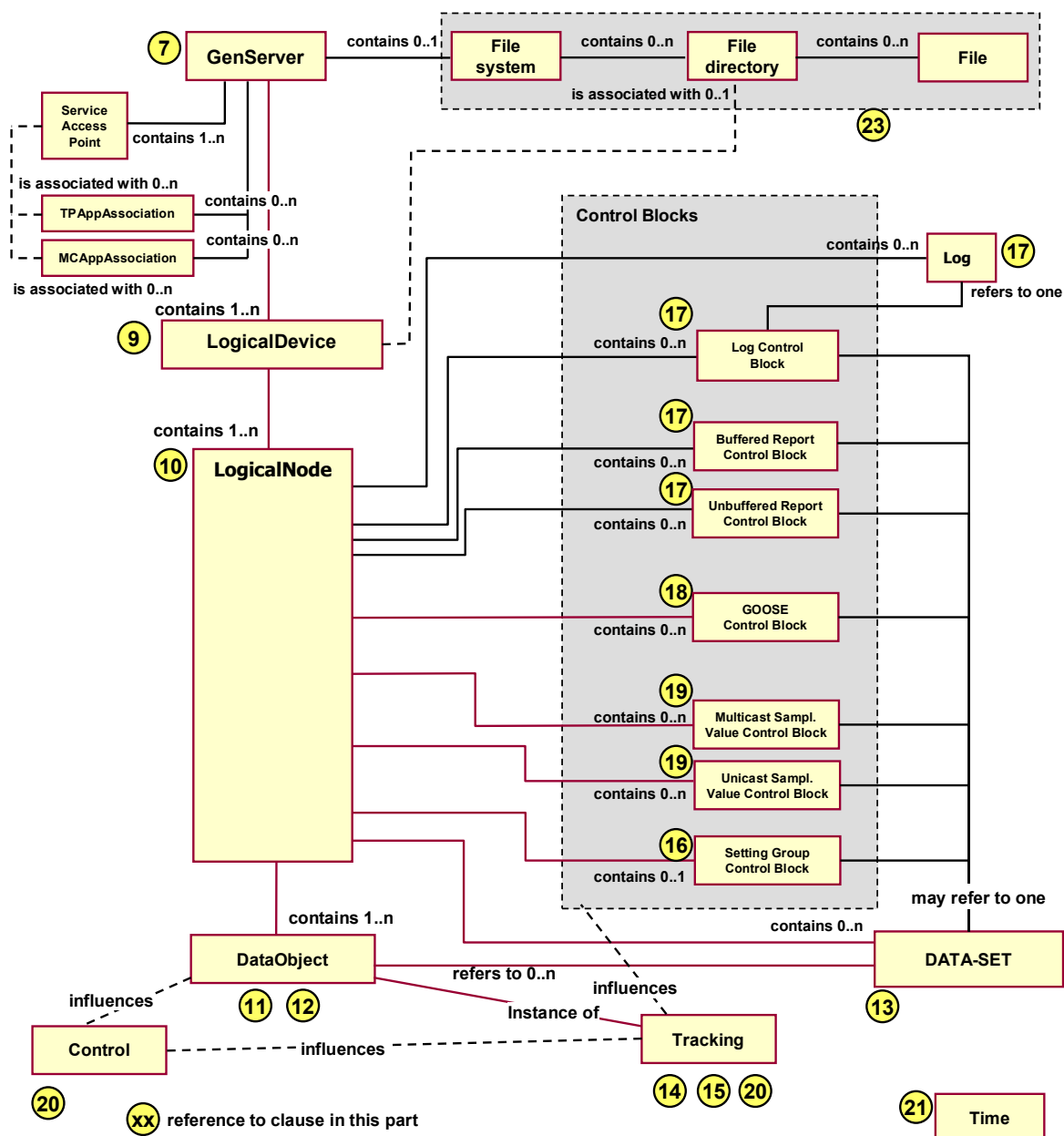


Figure 3 – Conceptual service model of the ACSI

NOTE 1 The numbers in the circles indicate the respective clauses in this part of IEC 61850.

NOTE 2 The class diagrams are conceptual. Details are defined in the respective clauses. Comprehensive diagrams are contained in IEC 61850-7-1.

The logical node is one of the major building blocks that have associations to most of the other information exchange models, for example, report control, log control, and setting control. In this part of IEC 61850, the generic logical node class is defined (**GenLogicalNode**).

NOTE 3 The class models and services are defined using an object-oriented approach allowing for the mapping of class models and services to different application layer and middle ware solutions.

The complete list of ACSI classes and their services is shown in Table 1.

Table 1 – ACSI model classes with related services

<u>GenServer model</u> GetServerDirectory <u>Association model</u> Associate Abort Release <u>GenLogicalDeviceClass model</u> GetLogicalDeviceDirectory <u>GenLogicalNodeClass model</u> GetLogicalNodeDirectory GetAllDataValues <u>GenDataObjectClass model</u> GetDataValues SetDataValues GetDataDirectory GetDataDefinition <u>DATA-SET model</u> GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory <u>SETTING-GROUP-CONTROL-BLOCK model</u> SelectActiveSG SelectEditSG SetEditSGValue ConfirmEditSGValues GetEditSGValue GetSGCBValues <u>REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK model</u> BUFFERED-REPORT-CONTROL-BLOCK: Report^a GetBRCBValues SetBRCBValues UNBUFFERED-REPORT-CONTROL-BLOCK: Report^a GetURCBValues SetURCBValues	LOG-CONTROL-BLOCK model: GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues <u>Generic substation event model – GSE</u> GOOSE SendGOOSEMessage^a GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues <u>Transmission of sampled values model</u> MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendMSVMessage^a GetMSVCBValues SetMSVCBValues UNICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendUSVMessage^a GetUSVCBValues SetUSVCBValues <u>Control model</u> Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate <u>Time and time synchronization</u> TimeSynchronization <u>FILE transfer model</u> GetFile SetFile DeleteFile GetFileAttributeValues
a) All the services for spontaneous sending are limited to one access point per control block instance.	

NOTE All services defined in this part of IEC 61850 operate on instances of classes only. The service GetDataValues for example operates on an instantiated data object class implemented in a real device. The service parameters in the abstract service tables and the definition of the services refer to those instances and not to the generic classes defined in this part of IEC 61850.

5.3.4 Relations between classes

The crucial relations of the meta model classes are shown in Figure 4. The figure shows the crucial building rule for data objects (the recursions). The abstract model in the ACSI uses the generic common data class model to define any hierarchical model of domain specific information. The class diagram uses two recursions: the GenCommonDataClass and the GenConstructedAttributeClass. These two recursions allow the definition of any common data class defined in IEC 61850-7-3.

Figure 4 shows some examples of the definitions of IEC 61850-7-3 (common data classes WYE, CMV; attributes like cVal, etc.) and IEC 61850-7-4 (logical node MMXU and data object PhV). The various types, names and identifiers (IDs) are shown in coloured boxes with the

part number indicated as "6" for IEC 61850-6, "7-3" for names and identifiers defined in IEC 61850-7-3, and "7-4" for logical node and data object names defined in IEC 61850-7-4.

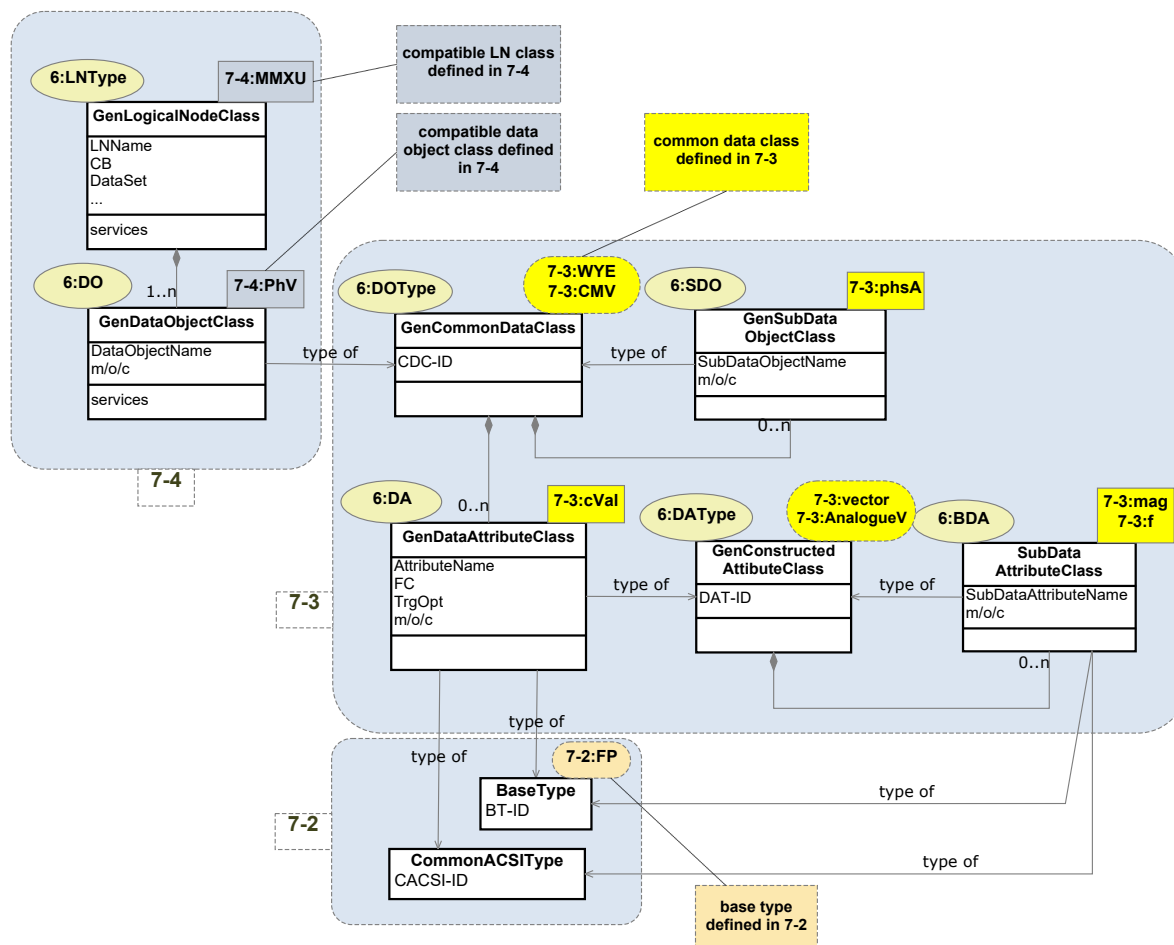


Figure 4 – Core of the conceptual meta model and relationship

The **GenCommonDataClass** is one of the crucial models to build the information models. The **GenCommonDataClass** model is used as a rule to define (build) common data classes (common for many domains like SPC for single point status or specific for a domain like WYE for electrical applications).

5.4 The domain type model

The domain type model of IEC 61850 defines lists of common data classes (CDC, IEC 61850-7-3), data objects (typed by common data classes) and logical node classes (IEC 61850-7-4) aggregating data objects. These classes are used to build data models for real IEDs.

IEC 61850-6 defines a method to describe the data instance model based on these classes, that can be used to describe the complete model implemented (programmed or configured) in a real IED. The data instance model is introduced in 5.5.

5.5 The data instance model

The data instance model describes instances of the classes defined in IEC 61850-7-x (see Figure 5). IEC 61850-6 defines by means of an XML schema (the SCL schema) a language to describe the configuration of IEDs. The SCL schema uses the element DOType to describe the common data class instantiation of a specific data object (DO element) in the logical node type (LNTType). IEC 61850-6 defines an IED element that has logical devices (LD) composed of

logical nodes (LN). The logical nodes are typed by instantiable LNTypes listed in the DataTypeTemplate section of an SCL document. Data objects in a LNTYPE become a data object instance (DOI) in the corresponding logical node.

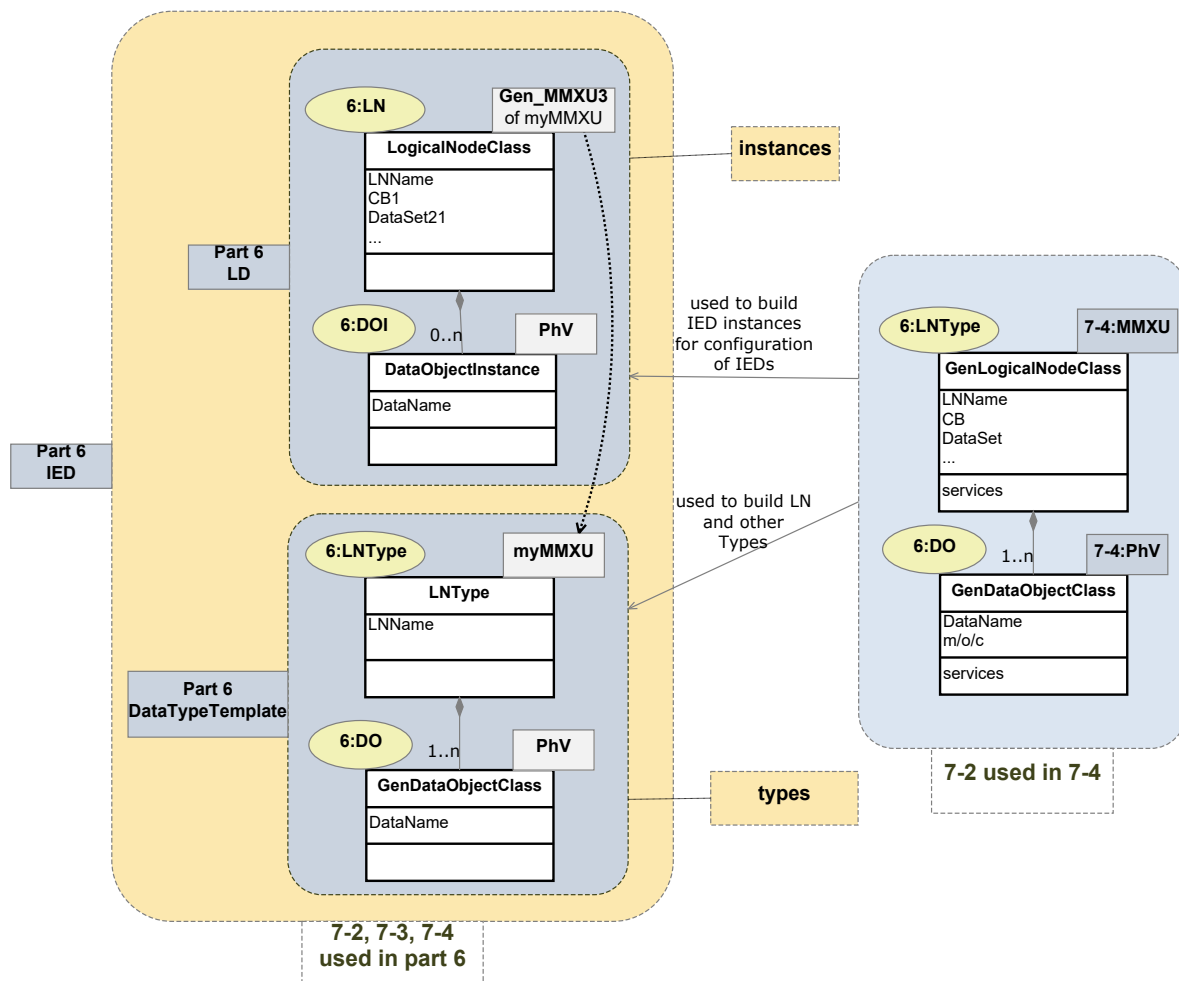


Figure 5 – Data instance model (conceptual)

NOTE The mapping of these instances to application layer protocols like MMS (manufacturing message specification, ISO 9506) are defined in the SCSMs. The logical device is mapped in MMS to a MMS domain class and the logical node and data objects are mapped to MMS NamedVariables as part of a domain.

6 TypeDefinitions

6.1 General

TypeDefinitions shall be composed of BasicTypes (6.1.1) and of CommonACSITypes (6.1.2).

6.1.1 BasicTypes

The **BasicTypes** shall be defined as listed in Table 2.

Table 2 – BasicTypes

BasicTypes			
Name	Value range	Remark	Used by
BOOLEAN	0,1 (false, true)		IEC 61850-7-3 IEC 61850-7-2
INT8	–128 to 127		IEC 61850-7-3 IEC 61850-7-2
INT16	–32 768 to 32 767		IEC 61850-7-3 IEC 61850-7-2
INT32	–2 147 483 648 to 2 147 483 647		IEC 61850-7-3 IEC 61850-7-2
INT64	–2**63 to (2**63) –1		IEC 61850-7-3
INT8U	Unsigned integer, 0 to 255		IEC 61850-7-3 IEC 61850-7-2
INT16U	Unsigned integer, 0 to 65 535		IEC 61850-7-3 IEC 61850-7-2
INT24U	Unsigned integer, 0 to 16 777 215	Only used for TimeStamp type	IEC 61850-7-2
INT32U	Unsigned integer, 0 to 4 294 967 295		IEC 61850-7-3 IEC 61850-7-2
FLOAT32	Range of values and precision as specified by IEEE 754 single- precision floating point		IEC 61850-7-3
ENUMERATED	Ordered set of values, defined where type is used. Values shall be assigned in the SCSMs.	Custom extensions are allowed	IEC 61850-7-3 IEC 61850-7-2
CODED ENUM	Ordered set of values, defined where type is used. Values shall be assigned in the SCSMs.	Custom extensions shall not be allowed. Type shall be mapped to an efficient encoding in a SCSM	IEC 61850-7-3 IEC 61850-7-2
OCTET STRING	Max. length shall be defined where type is used ^a	The NULL OCTET STRING is implemented by an empty OCTET STRING	IEC 61850-7-3 IEC 61850-7-2
VISIBLE STRING	Max. length shall be defined where type is used ^a	The NULL VISIBLE STRING is implemented by an empty VISIBLE STRING	IEC 61850-7-3 IEC 61850-7-2
UNICODE STRING	Unicode coding is defined in the SCSM. Max. length shall be defined where type is used ^a	The NULL UNICODE STRING is implemented by an empty UNICODE STRING	IEC 61850-7-3
Currency	A currency identification code based on ISO 4217 3-character currency code. The concrete coding shall be defined by the SCSMs.		IEC 61850-7-3
NOTE The data type INT128 is deprecated and replaced by INT64.			
^a The length suffix shall have the format "...STRINGnn" where "nn" is the length in characters.			

6.1.2 CommonACSTypes

6.1.2.1 General

The **CommonACSTypes** shall be used for the attribute definitions of the classes (for example, in report control blocks) defined in this part of IEC 61850. The **CommonACSTypes** may also be used in the application models defined in IEC 61850-7-3 and IEC 61850-7-4.

The following types are defined:

- ObjectName (see 6.1.2.2),
- ObjectReference (see 6.1.2.3),
- PHYCOMADDR type (see 6.1.2.4),
- ARRAY type (see 6.1.2.5),
- ServiceError type (see 6.1.2.6),
- EntryID type (see 6.1.2.7),
- Packed list type (see 6.1.2.8),
- TimeStamp type (see 6.1.2.9),
- EntryTime type (see 6.1.2.10),
- TriggerConditions type (see 6.1.2.11),
- ReasonCode (ReasonForInclusion) (see 6.1.2.12).

6.1.2.2 ObjectName

The **ObjectName** shall define a unique instance name among instances of a class owned by the same parent class with a type as defined in Table 3.

Table 3 – ObjectName type

ObjectName type			
Attribute name	Attribute type	Value/value range/explanation	Used by
ObjectName	VISIBLE STRING64	Name of an instance of a class of a single hierarchy level	IEC 61850-7-4 IEC 61850-7-3 IEC 61850-7-2

The constraints defined in Clause 22 on the use of the type ObjectReference shall be applied.

6.1.2.3 ObjectReference

Instances of classes in the hierarchical information model (ACSI class hierarchy of logical device, logical node, data, data attributes) shall be constructed by the concatenation of all instance names comprising the whole path-name of an instance of a class that identifies the instance uniquely. The type of the **ObjectReference** shall be as defined in Table 4.

Table 4 – ObjectReference type

ObjectReference type			
Attribute name	Attribute type	Value/value range/explanation	Used by
ObjectReference	VISIBLE STRING129	ObjectReference comprises the whole path-name of an instance of a class that identifies the instance uniquely. The ObjectReference shall be composed of two parts: up to 64 characters for the LD name followed by one separator "/" followed by up to 64 characters for the path below the LD name. The NULL ObjectReference is an empty ObjectReference (i.e. empty VISIBLE STRING129)	IEC 61850-7-3 IEC 61850-7-2

The **ObjectReference** syntax shall be:

LDName/LNName[.Name[. ...]]

- The "/" shall separate the instance name of a logical device (**LDName**) from the name of an instance of a logical node (**LNName**).
- The "." shall separate the further names in the hierarchy.
- The "[.]" indicates an option.
- The "[. ...]" indicates further names of recursively nested definitions.
- The "(...)" shall indicate an array element

NOTE In any case where the context of the text provides sufficient information that an instance of a class is meant, the term "instance of" is not used.

The constraints defined in Clause 22 on the use of the type **ObjectReference** shall be applied.

6.1.2.4 PHYCOMADDR type

The type **phycomaddr** shall represent a physical communication address (for example media access address, priority, and other information) as defined by a SCSM.

6.1.2.5 ARRAY type

The type **array** shall be defined as follows:

ARRAY 0..m OF p

with $m \geq 0$

p = **GenCommonDataClass** that does not contain an **array** type or
GenDataAttributeClass or
GenConstructedAttributeClass or
TypeDefinitions (except **ARRAY** type)

shall represent a list of elements numbered from 0 to "m". The type of the elements shall be as specified by "p".

6.1.2.6 ServiceError type

The service error code for negative service responses (originated within the server) shall be as specified in Table 5.

Table 5 – ServiceError type

ServiceError type definition			
Attribute name	Attribute type	Value /value range/explanation	Used by
ServiceError	ENUMERATED	instance-not-available instance-in-use access-violation access-not-allowed-in-current-state parameter-value-inappropriate parameter-value-inconsistent class-not-supported instance-locked-by-other-client control-must-be-selected type-conflict failed-due-to-communications-constraint failed-due-to-server-constraint no-error	IEC 61850-7-2

Additional **ServiceError** values for negative service responses (originated in the application, for example, additional cause diagnosis for control-related services) shall be as specified in the appropriate service models.

NOTE The **ServiceError** may be extended by an SCSM and the application layer referenced by an SCSM.

6.1.2.7 EntryID type

The type **EntryID** shall represent an arbitrary OCTET STRING used to identify an entry in a sequence of events such as a log or a buffered report as specified by an SCSM.

NOTE 1 The **EntryID** (handle) allows a client to re-synchronize, for example, with the sequence of the events stored in the IED. The syntax of the **EntryID** value is a local issue outside the scope of this standard. However, the NULL entryID used in the standard must be the OCTET STRING whose octets have all the value 0 (zero).

NOTE 2 The **EntryID** is used in this part of IEC 61850.

6.1.2.8 Packed list type

The **packed list** type shall be as defined in Table 6.

Table 6 – PACKED-LIST type

PACKED-LIST type definition			
Name	Value range	Remark	Used by
PACKED LIST	Ordered list of types; defined where type is used	Any value inside a PACKED LIST shall be mapped to an efficient encoding in a SCSM. No access to individual members of the list is required	IEC 61850-7-3 IEC 61850-7-2

6.1.2.9 TimeStamp type

6.1.2.9.1 General

The relation between a time stamp value, the synchronization of an internal time with an external time source (for example, UTC time), and other time-model-related information are defined in Clause 21.

NOTE 1 The TimeStamp type relies on requirements specified in Clause 21. The reader should first read that clause. The presentation of the TimeStamp is defined in the SCSMs.

NOTE 2 The TimeStamp is used in this part of IEC 61850 and in IEC 61850-7-3.

6.1.2.9.2 TimeStamp syntax

The **TimeStamp** type shall represent a UTC time with the epoch of midnight (00:00:00) of 1970-01-01 specified in Table 7.

Table 7 – TimeStamp type

TimeStamp type definition			
Attribute name	Attribute type	Value/value range/explanation	M/O
SecondSinceEpoch	INT32U	(0...MAX)	M
FractionOfSecond	INT24U	Value = SUM from i=0 to 23 of $b_i \cdot 2^{32-i-1}$; Order = $b_0, b_1, b_2, b_3, \dots$	M
TimeQuality	TimeQuality		M

6.1.2.9.3 TimeStamp attributes

6.1.2.9.3.1 SecondSinceEpoch

The **SecondSinceEpoch** shall be the interval in seconds continuously counted from the epoch 1970-01-01 00:00:00 UTC.

6.1.2.9.3.2 FractionOfSecond

The attribute **FractionOfSecond** shall be the fraction of the current second when the value of the **TimeStamp** has been determined. The fraction of second shall be calculated as (SUM from $i = 0$ to 23 of $b_i \cdot 2^{-(i+1)}$ s).

NOTE 1 The resolution is the smallest unit by which the time stamp is updated. The 24 bits of the integer provides 1 out of 16777216 counts as the smallest unit; calculated by $1/2^{24}$ which equals approximately 60 ns.

NOTE 2 The resolution of a time stamp may be $1/2^{21}$ (= 0,5 s) if only the first bit is used; or may be $1/2^{22}$ (= 0,25 s) if the first two bits are used; or may be approximately 60 ns if all 24 bits are used. The resolution provided by an IED is outside the scope of this standard.

6.1.2.9.3.3 TimeQuality

The **TimeQuality** shall provide information about the time source of the sending IED as listed in Table 8.

Table 8 – TimeQuality definition

TimeQuality definition			
Attribute name	Attribute type	Value/Value range/explanation	M/O
	PACKED LIST		
LeapSecondsKnown	BOOLEAN		M
ClockFailure	BOOLEAN		M
ClockNotSynchronized	BOOLEAN		M
TimeAccuracy	CODED ENUM	Number of significant bits in the FractionOfSecond: Minimum time interval shall be: 2^{-n}	M

LeapSecondsKnown: The value TRUE of the attribute **LeapSecondsKnown** shall indicate that the value for SecondSinceEpoch contains all leap seconds occurred. If it is FALSE, then the value does not take into account the leap seconds that occurred before the initialization of the time source of the device. Instead the seconds since start of the epoch are calculated from the current date assuming a constant day length of 86 400 s.

NOTE If a UTC time master clock is used and accessible, LeapSecondsKnown should always be true.

ClockFailure: The attribute **clockFailure** shall indicate that the time source of the sending device is unreliable. When ClockFailure is set, the value of the TimeStamp shall be ignored.

ClockNotSynchronized: When set, the attribute **clockNotSynchronized** shall indicate that the time source of the sending device is not synchronized with the external UTC time; otherwise **clockNotSynchronized** is FALSE.

TimeAccuracy: The attribute **TimeAccuracy** shall represent the time accuracy class of the time respective time stamp relative to the external UTC time. The exact meaning depends on the usage of the time stamp respective on the time stamped object. The **timeAccuracy** classes shall represent the number of significant bits in the **FractionOfSecond**.

The values of n shall be as listed in Table 9.

NOTE The **TimeAccuracy** meets the requirements specified in IEC 61850-5 for the selected values of n.

Table 9 – TimeAccuracy

n	Resulting TimeAccuracy (2 ⁿ –n)	Corresponding time performance class defined in IEC 61850-5	
31	–	–	unspecified
7	approx. 7,8 ms	10	ms (performance class T0)
10	approx. 0,9 ms	1	ms (performance class T1)
14	approx. 61 µs	100	µs (performance class T2)
16	approx. 15 µs	25	µs (performance class T3)
18	approx. 3,8 µs	4	µs (performance class T4)
20	approx. 0,9 µs	1	µs (performance class T5)

The NULL TimeStamp is a TimeStamp whose fields are all set to 0 (zero).

6.1.2.10 EntryTime type

The type **EntryTime** shall represent the time and date as applied internally for the communication, reporting, logging, and subsystem as specified by a SCSM.

The time base for **EntryTime** shall be GMT. The epoch for EntryTime shall be 01. January 1984 (MJD 40 587).

NOTE 1 The TimeStamp type is used for common data classes in IEC 61850-7-3 and definition of compatible data object classes in IEC 61850-7-4. The **EntryTime** type may or may not be the same as **TimeStamp** in a SCSM.

NOTE 2 The EntryTime is used in this part of IEC 61850.

6.1.2.11 TriggerConditions type

The **TriggerConditions** type shall represent the trigger conditions used to trigger processing reports (see Table 10).

Table 10 – TriggerConditions type

TriggerConditions type				
Attribute name		Attribute type	Value / Value range	M/O/C
		PACKED LIST		M
	data-change	BOOLEAN	See Clause 17	M
	quality-change	BOOLEAN	See Clause 17	M
	data-update	BOOLEAN	See Clause 17	M
	integrity	BOOLEAN	See Clause 17	M
	general-interrogation	BOOLEAN	See Clause 17	M

NOTE Details on the use of **TriggerConditions** are defined in Clause 17.

6.1.2.12 ReasonCode (ReasonForInclusion)

The values conveyed by ReasonForInclusion shall be a PACKEDLIST as defined in Table 11.

Table 11 – ReasonForInclusion

ReasonForInclusion			
Attribute Name	Attribute Type	Value/Value Range	M/O/C
	PACKEDLIST		
data-change	BOOLEAN	may only be TRUE if TrgOp.dchg = TRUE	M
quality-change	BOOLEAN	may only be TRUE if TrgOp.qchg = TRUE	M
data-update	BOOLEAN	may only be TRUE if TrgOp.dupd = TRUE	M
integrity	BOOLEAN	may only be TRUE if IntgPd is non-zero and TrgOp.integrity = True	M
general-interrogation	BOOLEAN	may only be TRUE if there has been a SetBRCBValues of GI=TRUE and TrgOp.general-interrogation=TRUE	M
application-trigger	BOOLEAN	may only be TRUE if the trigger comes from an application function	AC_LOG_M
AC_LOG_M: The attribute shall only be present when ReasonForInclusion is used within the scope of LOG.			

General-interrogation, integrity and application-trigger reasons for inclusions are mutually exclusive of all other reasons for inclusion (see 17.2.3.2.3.5 and 17.3.3.2.7.3).

7 GenServerClass model

7.1 GenServerClass definition

7.1.1 GenServerClass syntax

The **GenServerClass** shall represent the externally visible behaviour of a device. The **GenServerClass** shall be a composition as defined in Table 12.

NOTE 1 For simple devices, the server may comprise just one logical device with the GOOSE control model with no other service.

Table 12 – GenServerClass definition

GenSERVER class		
Attribute name	Attribute type	Value/value range/explanation
ServiceAccessPoint [1..n]	(*)	(*) Type is SCSM specific
LogicalDevice [1..n]	GenLogicalDeviceClass	
FileSystem [0..1]	FILE-SYSTEM	
TPAppAssociation [0..n]	TWO-PARTY-APPLICATION-ASSOCIATION	
MCAAppAssociation [0..n]	MULTICAST-APPLICATION-ASSOCIATION	
Services		
GetServerDirectory		

NOTE 2 The server's relationship to the underlying communication system and the concrete implementation depend on the SCSM (specific communication service mapping, see IEC 61850-8-x and IEC 61850-9-x) used. Network management (as part of an SCSM), device management, and system management are outside the scope of IEC 61850-7-2.

7.1.2 GenServerClass attributes

7.1.2.1 ServiceAccessPoint [1..n]

The attribute **ServiceAccessPoint** shall identify a server within the scope of a subnetwork.

NOTE The **ServiceAccessPoint** is an abstraction of an address used to identify the server and a profile of the underlying SCSM. The type depends on the SCSM and should be defined there. A specific **ServiceAccessPoint** is required by most services to address a server. Nevertheless, it has not been included explicitly in the service parameter tables throughout this part of IEC 61850.

7.1.2.2 LogicalDevice [1..n]

The attribute **LogicalDevice** shall identify a logical device that is contained in a **GenServer**.

7.1.2.3 FileSystem [0..n]

The attribute **FileSystem** shall identify a File System contained in a **GenServer**.

7.1.2.4 TPAAppAssociation [0..n] – two-party application association

The attribute **TPAppAssociation** shall identify a client with which a server maintains a two-party application association.

NOTE Details can be found in Clause 8.

7.1.2.5 MCAAppAssociation [0..n] – multicast application association

The attribute **MCAAppAssociation** shall identify a subscriber with which a server (publisher) maintains a multicast application association.

NOTE Details can be found in Clause 8.

7.2 Server class services

7.2.1 Overview of directory and GetDefinition services

To support self-description of a device, several GetXXDirectory and GetXXDefinition services as shown in Figure 6 are specified in this part of IEC 61850.

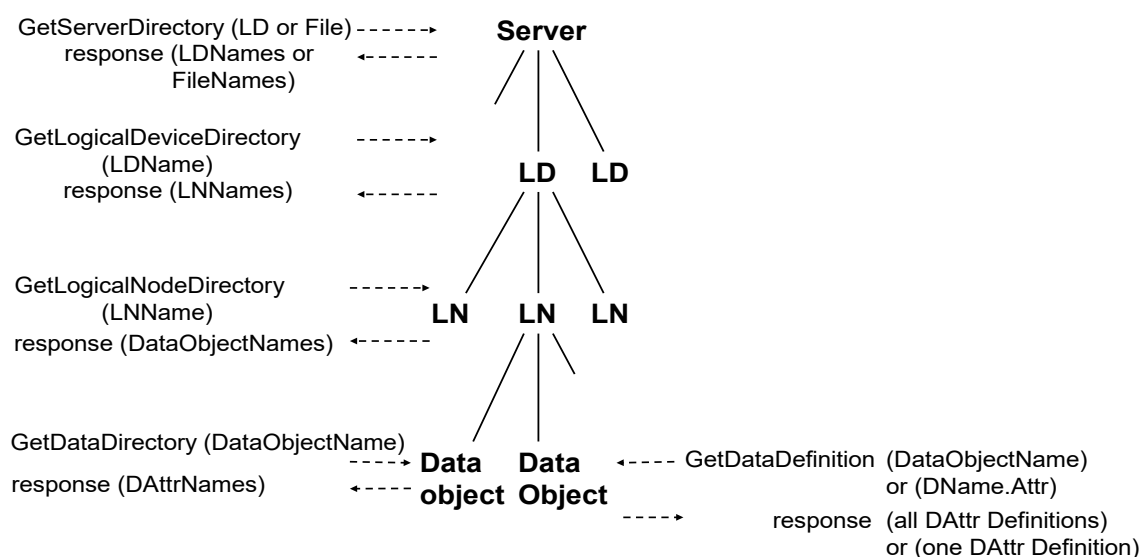


Figure 6 – Overview about GetDirectory and GetDefinition services

A client shall use these services to retrieve the definition of the complete hierarchy – as well as the definition of all accessible information – and of all instances of all underlying classes in a given server.

7.2.2 GetServerDirectory

7.2.2.1 GetServerDirectory parameter table

A client shall use the **GetServerDirectory** service to retrieve a list of the names of all logical devices and file systems made visible and thus accessible to the requesting client by the addressed server.

Parameter name
Request
ObjectClass
Response+
Reference [0..n]
Response–
ServiceError

7.2.2.2 Request

7.2.2.2.1 ObjectClass

The parameter **ObjectClass** shall contain an identification of the selected class. The client shall select one of the following classes:

- **logical-device**
- **file –system**

NOTE The syntax of the value of an **ObjectClass** is defined in an SCSM.

7.2.2.3 Response+

The parameter **Response+** shall indicate that the service request succeeded. A successful result shall return the following parameter.

7.2.2.3.1 Reference [0..n]

The parameter **Reference** shall contain the ObjectReference of the logical devices (when ObjectClass is set to LOGICAL-DEVICE) or shall contain the file name(s) present in the file system (when ObjectClass is set to FILE-SYSTEM).

NOTE The **FileName** type is a VISIBLE STRING255 and is defined in 23.1.1.

7.2.2.4 Response–

The parameter **Response–** shall indicate that the service request failed. The appropriate ServiceError shall be returned, for example, failed-due-to-server-constraint, parameter-value-inconsistent, or parameter-value-inappropriate.

8 Application association model

8.1 Introduction

The application association model consists of provisions on how the communication between the various types of devices is achieved. The model comprises

- class definitions of associations (two-party and multicast); and
- access control concepts (how to restrict access to instances in a server).

The security requirements for the restriction of access to the data in a server are defined in IEC 61850-5.

NOTE Security implementations are defined in the SCSMs.

8.2 Concept of application associations

The application association model defines

- the services provided for managing associations between client and server (two-party application association); and
- the services provided for managing associations for multicast messaging (for example, GOOSE and transmission of sampled values).

The **two-party application association** class shall convey service requests and responses (thus transferring unconfirmed and confirmed services). The **multicast application association** class shall be capable of conveying unconfirmed services (in one direction only).

Application associations provide a mechanism for controlling the access to the instances of a device (access control).

NOTE The details of an application association model are defined in the SCSMs. The following descriptions provide a conceptual model of the application associations between devices.

8.3 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model

8.3.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition

8.3.1.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class syntax

A two-party application association type shall provide a bi-directional connection-oriented information exchange. The application associations shall be reliable and the information flow shall be controlled end to end. Reliable means that the connection on which the application association relies provides measures to notify reasons for non-deliverance of information in due time. End-to-end flow control means that sources of information do not send more information than the destination can buffer.

The services for associate, data exchange, and association release of the two-party application association class is depicted in Figure 7.

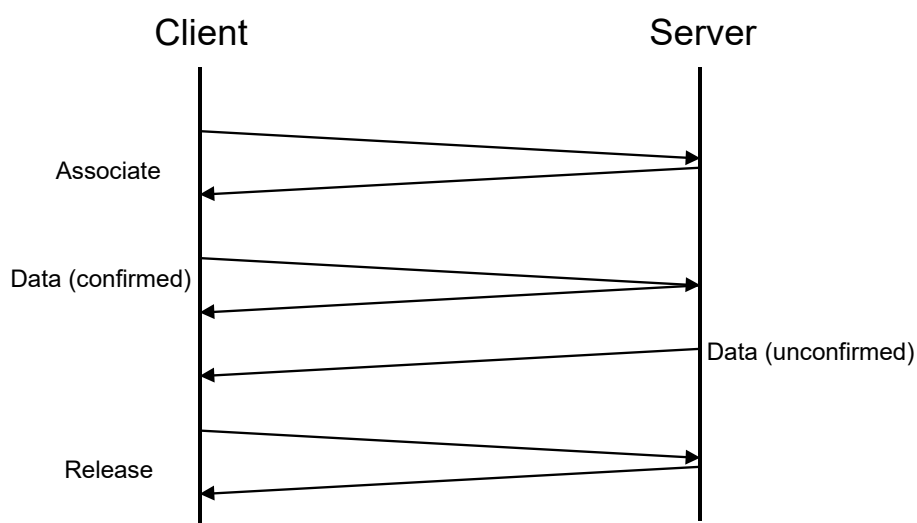


Figure 7 – Normal operation

The abort service for the two-party application association class is depicted in Figure 8.

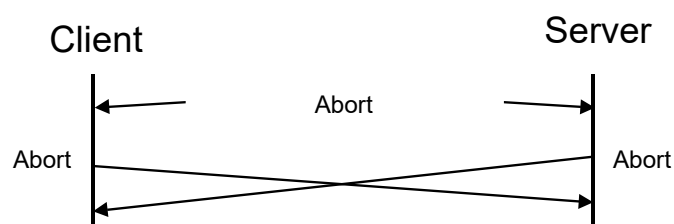


Figure 8 – Aborting association

The **two-party-application-association** (TPAA) class shall be defined as in Table 13.

Table 13 – TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition

TWO-PARTY-APPLICATION-ASSOCIATION class		
Attribute name	Attribute type	Value/value range/explanation
AssociationId	(*)	(*) Type is SCSM specific
AuthenticationParameter	(*)	(*) Type is SCSM specific
Services Associate Abort Release Additional services that make use of the TWO-PARTY-APPLICATION-ASSOCIATION shall be as indicated in Table A.3 of Clause A.4 (in column Asso. marked as "TP").		

8.3.1.2 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class attributes

8.3.1.2.1 AssociationId

The attribute **AssociationId** shall define the identification used to identify the application associations.

NOTE The type of the **AssociationId** is defined in the SCSMs and it may be exchanged in an SCSM or be used locally only.

8.3.1.2.2 AuthenticationParameter

The attribute **authenticationParameter** shall represent the information required to grant permission to access instances of a specific access view to a server.

NOTE A minimum set of parameters is user identification and password. The details are defined in the SCSMs.

8.3.2 Two-party application association services

8.3.2.1 Overview

For **two-party-application-association**, the following services are defined.

Service	Description
Associate	Establish an association
Abort	Abort an association
Release	Release an association

8.3.2.2 Associate

8.3.2.2.1 Associate parameter

A client shall use the **Associate** service to establish an application association of type two-party with a specific server.

Parameter name
Request
ServerAccessPointReference
AuthenticationParameter
Response+
AssociationId
Result
Response–
ServiceError

8.3.2.2.2 Request

8.3.2.2.2.1 ServerAccessPointReference

The parameter **ServeAccessPointReference** shall identify the server, with which the application association shall be established.

8.3.2.2.2.2 AuthenticationParameter

This parameter **AuthenticationParameter** shall define the **authenticationParameter** for this application association to be opened. If an **authenticationParameter** does not match with a valid parameter, the service request shall be rejected and an appropriate reason shall be returned.

NOTE The type of the **authenticationParameter** is defined in the SCSM.

8.3.2.2.3 Response+

AssociationId

The parameter **AssociationId** may be used to differentiate the application associations.

NOTE The **AssociationId** may be exchanged in a response+ message of an SCSM or be used locally only.

8.3.2.2.4 Result

The parameter **Result** shall indicate if the establishment of the application association was successful or not.

8.3.2.2.5 Response–

The parameter **Response–** shall indicate that the service request failed. The appropriate **ServiceError** shall be returned.

8.3.2.3 Abort

8.3.2.3.1 Abort parameter

The service **Abort** shall be used to abruptly disconnect a specific application association between a client and a server. Abrupt means that all service requests issued shall be discarded – no further service shall be processed.

Parameter name
Request
AssociationId
Reason
Indication
AssociationId
Reason

8.3.2.3.2 Request

8.3.2.3.2.1 AssociationId

The parameter **AssociationId** shall define the association to be aborted. The indication may be issued by the underlying layer (locally or remotely) or it may be sent from remote user of the association.

8.3.2.3.2.2 Reason

The parameter **Reason** shall define the reason why the association has been aborted. The reason may be provided by the underlying layer (locally or remotely) or it may be sent from remote user of the association.

8.3.2.3.3 Indication

8.3.2.3.3.1 AssociationId

The parameter **AssociationId** shall define the association that has been aborted.

8.3.2.3.3.2 Reason

The parameter **Reason** shall define the reason for abrupt termination the application association.

8.3.2.4 Release

8.3.2.4.1 Release parameter

The service **Release** shall be used to gracefully disconnect a specific application association between a client and a server. Graceful means that all service requests issued shall be completed before termination. New request shall not be issued after disconnect initiation.

Parameter name
Request
AssociationId
Response+
AssociationId
Result
Response–
ServiceError

8.3.2.4.2 Request

AssociationId

The parameter **AssociationId** shall define the association to be terminated.

8.3.2.4.3 Response+

8.3.2.4.3.1 AssociationId

The parameter **AssociationId** shall define the association that has been terminated.

8.3.2.4.3.2 Result

The parameter **Result** shall indicate, if the termination of the application association was successful.

8.3.2.4.4 Response–

The parameter **Response–** shall indicate that the service request failed. The appropriate ServiceError shall be returned, for example, instance-not-available, parameter-value-inappropriate, parameter-value-inconsistent, failed-due-to-communications-constraint, or failed-due-to-server-constraint.

In the case of a **Release** requested before the completion of (a) pending service(s), the Server shall answer with **Response–**. The application association shall not be terminated.

8.4 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class

8.4.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition

A multicast application association type shall provide a unidirectional information exchange. Multicast information exchange shall be provided between one source (publisher) and one or many destinations (subscriber). Unidirectional information exchange shall provide sufficient information for the receivers to uniquely interpret the context in which the exchange shall be processed. The subscriber shall be capable to detect loss and duplication of information received. The receiver shall notify the loss of information to its user and shall discard duplicated information.

NOTE The possible restriction of multicast messages to be exchanged on a single subnet or sent through routers is an issue to be defined in an SCSM.

The multicast application association class is depicted in Figure 9.

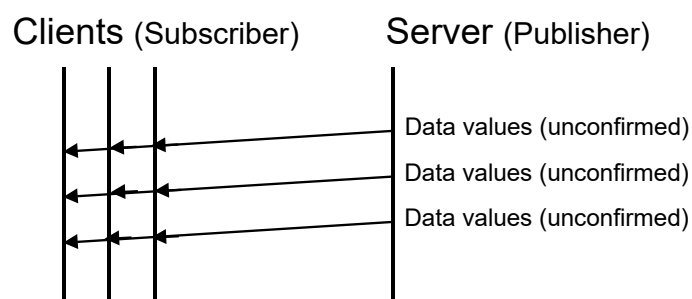


Figure 9 – Principle of multicast application association

The **multicast-application-association (MCAA)** shall be as defined in Table 14.

Table 14 – MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition

MULTICAST-APPLICATION-ASSOCIATION class		
Attribute name	Attribute type	Value/value range/explanation
AuthenticationParameter	(*)	(*) Type is SCSM specific
Services Services that make use of the MULTICAST-APPLICATION-ASSOCIATION shall be as indicated in Table A.3 of Clause A.4 (in column Asso. marked as "MC").		

8.4.2 MULTICAST-Application-association (MCAA) class attributes

8.4.2.1 AuthenticationParameter

The **authenticationParameter** shall represent the information required to grant permission to access instances of a specific access view to a client.

Each multicast service shall provide a service parameter that specifies the **authenticationParameter** for this data exchange. If an **authenticationParameter** does not match with a valid parameter, the service request shall be rejected by the receiving device.

NOTE 1 The type of the **authenticationParameter** is defined in the SCSM.

NOTE 2 Each exchange of information using multicast services can be understood as an "associate message" that carries association parameters and data. The "application association" ceases as soon as the service has been processed.

9 GenLogicalDeviceClass model

9.1 GenLogicalDeviceClass definition

9.1.1 GenLogicalDeviceClass syntax

The **GenLogicalDeviceClass** (**GenLD**) shall be a composition of **GenLogicalNodeClass** as defined in Table 15.

NOTE A **GenLogicalDeviceClass** can be used simply as a container of a group of **GenLogicalNodeClass** or as a device that functions as a gateway or proxy. Details on the use of **GenLogicalDeviceClass** can be found in IEC 61850-7-1.

Table 15 – GenLogicalDeviceClass (GenLD) class definition

GenLOGICAL-DEVICE class		
Attribute name	Attribute type	Value/value range/explanation
LDName	ObjectName	Instance name of an instance of GenLogicalDeviceClass
LogicalNode [1..n]	GenLogicalNodeClass	IEC 61850-7-4 specifies specialized classes of GenLogicalNodeClass
Services GetLogicalDeviceDirectory		

9.1.2 GenLogicalDeviceClass attributes

9.1.2.1 LDName – logical device name

The attribute **LDName** shall unambiguously identify an instance of a logical device within the scope of a subnetwork.

9.1.2.2 LogicalNode [1..n]

The attribute **LogicalNode** [1..n] shall be a list of all logical nodes of that are contained in a **GenLogicalDeviceClass**.

Each **GenLogicalDeviceClass** shall have one and only one **logical-node-zero (LLN0)** and it may have no or several other **LogicalNodes**.

NOTE The details of **LLN0** and other logical node classes are defined in IEC 61850-7-4 for utility automation applications.

9.2 GenLogicalDeviceClass services

9.2.1 GetLogicalDeviceDirectory

9.2.1.1 GetLogicalDeviceDirectory parameter table

A client shall use the **GetLogicalDeviceDirectory** service to retrieve the list of the ObjectReferences of all logical nodes made visible and thus accessible to the requesting client by the referenced logical device.

Parameter name
Request
LDName
Response+
LNReference [1..n]
Response–
ServiceError

9.2.1.2 Request

9.2.1.2.1 LDname – logical device object name

The parameter **LDName** shall contain the object name of a logical device.

9.2.1.3 Response+

The parameter **Response+** shall indicate that the service request succeeded. A successful result shall return the following parameter.

The parameter **LNReference [1..n]** shall contain ObjectReferences of all logical nodes from the referenced logical device. At least one reference shall be returned according to IEC 61850-7-4: for the logical node zero (**LLNO**).

9.2.1.4 Response–

The parameter **Response–** shall indicate that the service request failed. The appropriate ServiceError shall be returned, for example if the LogicalDevice specified with the LDName of the GetLogicalDevice request does not exist in the server.

10 GenLogicalNodeClass model

10.1 GenLogicalNodeClass definition

10.1.1 GenLogicalNodeClass diagram

The model of the GenLogicalNodeClass is depicted in Figure 10. The GenLogicalNodeClass (for example a class MMXU – measurements of a 3-phase electrical system) comprises several attributes and services. It is an aggregation of data objects (for example, PhV – phase voltages).

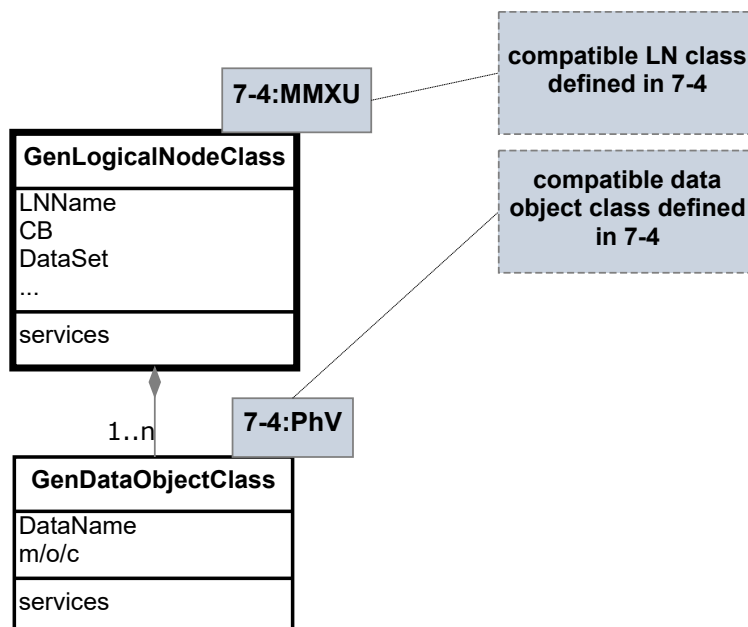


Figure 10 – Basic conceptual model of the GenLogicalNodeClass

The GenLogicalNodeClass details are defined in the following clauses.

10.1.2 GenLogicalNodeClass syntax

The **GenLogicalNodeClass** shall be a composition of **DataObjects**, **DATA-SET**, **BRCB**, **URCB**, **LCB**, **LOG**, **SGCB**, **GoCB**, **GSCB**, **MSVCB**, and **USVCB** as defined in Table 16.

Table 16 – GenLogicalNodeClass definition

GenLogicalNodeClass		
Attribute name	Attribute type	Explanation
LNNName	ObjectName	Instance name of an instance of LOGICAL-NODE
LNRef	ObjectReference	Path-name of an instance of LOGICAL-NODE
DataObject [1..n]	GenCommonDataClass	
DataSet [0..n]	DATA-SET	
BufferedReportControlBlock [0..n]	BRCB	
UnbufferedReportControlBlock [0..n]	URCB	
The following attributes shall only be available if their support is explicitly stated in the definition of a compatible LN class, for example, in IEC 61850-7-4.		
SettingGroupControlBlock [0..1]	SGCB	
Log [0..n]	LOG	
LogControlBlock [0..n]	LCB	
GOOSEControlBlock [0..n]	GoCB	
MulticastSampledValueControlBlock [0..n]	MSVCB	
UnicastSampledValueControlBlock [0..n]	USVCB	
Services GetLogicalNodeDirectory GetAllDataValues		
NOTE IEC 61850-7-4 defines specialized logical node classes – the compatible logical node classes, for example, XCBB representing circuit-breakers.		

The definitions of **GenDataObjectClasses** for the utility automation domain are refined by the definition of specific data objects in IEC 61850-7-4. The definitions in IEC 61850-7-4 (and IEC 61850-7-3 for the common data classes) shall be taken into account to get the comprehensive definition of utility automation-domain-specific logical nodes.

NOTE IEC 61850-7-4 defines further attributes for logical nodes; for example, the mode (behaviour: ON, TEST, TEST-BLOCKED etc.) of the utility automation-specific logical node is defined in IEC 61850-7-4. The state model of a logical node is modelled as a specific data object (named **Mod**).

10.1.3 GenLogicalNodeClass attributes

10.1.3.1 LNName – Logical node name

The attribute **LNName** shall unambiguously identify a logical node within the scope of a logical device.

10.1.3.2 LNRef – Logical node ObjectReference

The attribute **LNRef** shall be the unique path-name of a logical node.

The ObjectReference **LNRef** shall be:

LDName/LNName

10.1.3.3 DataObject [1..n]

The attribute **DataObject** shall be a data object that is contained in a logical node. The **DataObject** details are defined in Clause 11.

NOTE IEC 61850-7-4 defines standardized data object classes.

10.1.3.4 DataSet [0..n]

The attribute **DataSet** shall be a list of all data sets that are contained in a logical node. The **DataSet** details are defined in Clause 13.

10.1.3.5 BufferedReportControlBlock [0..n]

The attribute **BufferedReportControlBlock** shall be a list of all buffered report control blocks that are contained in a logical node. The **BufferedReportControlBlock** details are defined in 17.2.2.

10.1.3.6 UnbufferedReportControlBlock [0..n]

The attribute **UnbufferedReportControlBlock** shall be a list of all unbuffered report control blocks that are contained in a logical node. The **UnbufferedReportControlBlock** details are defined in 17.2.4.

10.1.3.7 Log [0..n]

The attribute **Log** shall be a list of all logs that are contained in a logical node. The **Log** details are defined in 17.3.

10.1.3.8 LogControlBlock [0..n]

The attribute **LogControlBlock** shall be a list of all log control blocks that are contained in a logical node. The **LogControlBlock** details are defined in 17.3.2.

10.1.3.9 SettingGroupControlBlock [0..1]

The attribute **SettingGroupControl** shall be the setting group control block that is contained in a logical node. The **SettingGroupControl** details are defined in Clause 16.

10.1.3.10 GOOSEControlBlock [0..n]

The attribute **GOOSEControlBlock** shall be a list of all GOOSE control blocks that are contained in a logical node. The **GOOSEControlBlock** details are defined in 18.2.

10.1.3.11 MulticastSampledValueControlBlock [0..n]

The attribute **MulticastSampledValueControlBlock** shall be a list of all multicast sampled value control blocks that are contained in a logical node. The **MulticastSampledValueControlBlock** details are defined in 19.2.

10.1.3.12 UnicastSampledValueControlBlock [0..n]

The attribute **UnicastSampledValueControlBlock** shall be a list of all unicast sampled value control blocks that are contained in a logical node. The **UnicastSampledValueControlBlock** details are defined in 19.3.

10.2 GenLogicalNodeClass services

10.2.1 Overview

For **GenLogicalNodeClass**, the following services are defined:

Service	Description
GetLogicalNodeDirectory	Retrieve ObjectReferences of a specific ACSI class contained in the LOGICAL-NODE
GetAllDataValues	Retrieve all DataAttribute values of all data object contained in a logical node

10.2.2 GetLogicalNodeDirectory

10.2.2.1 GetLogicalNodeDirectory parameter table

A client shall use the **GetLogicalNodeDirectory** service to retrieve a list of the ObjectReferences of all instances of a requested class made visible and thus accessible to the requesting client by the referenced logical node.

Parameter name
Request
LNReference
ACSIClass
Response+
InstanceName [0..n]
Response–
ServiceError

10.2.2.2 Request

10.2.2.2.1 LNReference

The parameter **LNReference** shall contain the ObjectReference of the logical node.

10.2.2.2.2 ACSIClass

The parameter **ACSIClass** shall contain an identification of the selected ACSI class model for which the ObjectReferences of all ACSI class models shall be returned.

The client shall select one identification for a class from the following ACSI class models:

Data object, DATA-SET, BRCB, URCB, LCB, LOG, SGCB, GoCB, GsCB, MSVCB, and USVCB.

NOTE GsCB is deprecated and kept only for backwards compatibility.

10.2.2.3 Response+

The parameter **Response+** shall indicate that the service request succeeded. A successful result shall return the following parameter.

InstanceName [0..n]

If the ACSI class model requested equals DATA-SET, BRCB, URCB, LCB, LOG, SGCB, GoCB, GsCB, MSVCB, or USVCB, the parameter **InstanceName [0..n]** shall contain all names of all instances of the requested class.

If the ACSI class model requested equals **Data object**, the parameter **InstanceName [0..n]** shall contain all names of the highest level of the data object instance.

NOTE If the data object instance contains a data object (for example data object PhV containing a data object phsA according to the CDC WYE), the name of the underlying data object may be retrieved by the service GetDataDirectory (11.4.4).

In the case where the referenced logical node does not contain the requested ACSI class, the server shall indicate that no object for the requested ACSI class model exists in this logical node.

10.2.2.4 Response–

The parameter **Response–** shall indicate that the service request failed. The appropriate ServiceError shall be returned.

10.2.3 GetAllDataValues

10.2.3.1 GetAllDataValues parameter table

A client shall use the **GetAllDataValues** service to retrieve all data attribute values (having the same **FunctionalConstraint**) of all data objects made visible and thus accessible to the requesting client by the referenced logical node.

Parameter name
Request
LNReference
FunctionalConstraint [0..1]
Response+
DataAttributeReference [1..n]
DataAttributeValue [1..n]
Response–
ServiceError

10.2.3.2 Request

10.2.3.2.1 LNReference

The parameter **LNReference** shall contain the ObjectReference of the logical node.

10.2.3.2.2 FunctionalConstraint [0..1]

The parameter **FunctionalConstraint (FC)** shall contain the functional constraint parameter (**FC**) to filter the respective data attributes of all data objects contained in the logical node. The **FC** shall be as defined in 12.3.3.2.

10.2.3.3 Response+

The parameter **Response+** shall indicate that the service request succeeded. A successful result shall return the following parameters.

10.2.3.3.1 DataAttributeReference [1..n]

The parameter **DataAttributeReference** shall contain the ObjectReference of a data attribute contained in the logical node that shall be returned according to the value of the **FunctionalConstraint** received in the request.

NOTE The ObjectReference **DataAttributeReference** is defined in 12.6.2.

10.2.3.3.2 DataAttributeValue [1..n]

The parameter **DataAttributeValue** shall contain the value of a data attribute of the data object contained in the referenced logical node. If the parameter **FunctionalConstraint** is present in the service request then only values of those data attributes that have the **FunctionalConstraint** as given in the service request shall be returned.

NOTE The syntax of the value of a data attribute is defined in an SCSM.

10.2.3.4 Response–

The parameter **Response–** shall indicate that the service request failed. The appropriate ServiceError shall be returned.