

DIN IEC 61850-7-2

ICS 33.040.40

Einsprüche bis 2008-04-30
Vorgesehen als Ersatz für
DIN EN 61850-7-2:2004-02

Entwurf

**Kommunikationsnetze und -systeme für die Automatisierung in der elektrischen Energieversorgung –
Teil 7-2: Abstrakte Schnittstelle für Kommunikationsdienste (ACSI)
(IEC 57/891/CD:2007), Text Englisch**

Communication networks and systems for power utility automation –
Part 7-2: Basic information and communication structure –
Abstract communication service interface (ACSI) (IEC 57/891/CD:2007), text English

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2008-02-18 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfes besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise als Datei per E-Mail an dke@din.de in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (Hausanschrift: Stresemannallee 15, 60596 Frankfurt am Main).

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevante Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 186 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Nationales Vorwort

Die englische Originalfassung des internationalen Dokuments IEC 57/891/CD:2007 „Communication networks and systems for power utility automation – Part 7-2: Basic information and communication structure – Abstract communication service interface (ACSI)“ (CD, en: Committee Draft) ist unverändert in diesen Norm-Entwurf übernommen worden.

Das internationale Dokument wurde vom TC 57 „Power systems management and associated information exchange“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet und den nationalen Komitees zur Stellungnahme vorgelegt.

Die IEC und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Da der Abstimmungszeitraum für einen FDIS bzw. Schluss-Entwurf prEN nur 2 Monate beträgt, und dann keine sachlichen Stellungnahmen mehr abgegeben werden können, sondern nur noch eine „JA/NEIN“-Entscheidung möglich ist, wobei eine „NEIN“-Entscheidung fundiert begründet werden muss, wird bereits der CD als DIN-Norm-Entwurf veröffentlicht, um die Stellungnahmen aus der Öffentlichkeit frühzeitig berücksichtigen zu können.

Für diesen vorliegenden Norm-Entwurf ist das nationale Arbeitsgremium K 952 „Netzleittechnik“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Änderungen

Gegenüber DIN EN 61850-7-2:2004-02 wurden folgende Änderungen vorgenommen:

- a) Klarstellung und Berichtigung
- b) Neue Dienste hinzugefügt
- c) IT-Sicherheit berücksichtigt

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

IEC 61850-7-2

Draft Ed2

Version: 61850-7-2_R1-00 Draft Ed2

CONTENTS

1		
2		
3	FOREWORD.....	9
4	INTRODUCTION.....	11
5	1 Scope.....	12
6	2 Normative references.....	12
7	3 Terms and definitions	13
8	4 Abbreviated terms.....	14
9	5 ACSI overview and basic concepts.....	16
10	5.1 General.....	16
11	5.2 Overview of basic information models.....	16
12	5.3 Overview of the other service models.....	17
13	5.4 Overview of ACSI services.....	20
14	5.5 Type definitions	21
15	5.5.1 General	21
16	5.5.2 Basic types.....	23
17	5.5.3 Common ACSI types.....	24
18	6 SERVER class model	29
19	6.1 SERVER class definition.....	29
20	6.1.1 SERVER class syntax	29
21	6.1.2 SERVER class attributes.....	30
22	6.2 Server class services.....	30
23	6.2.1 Overview of directory and GetDefinition services	30
24	6.2.2 GetServerDirectory	31
25	7 Application association model	32
26	7.1 Introduction	32
27	7.2 Concept of application associations	32
28	7.3 Access control.....	32
29	7.4 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model.....	33
30	7.4.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition	33
31	7.4.2 Two-party application association services	35
32	7.5 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class	38
33	7.5.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition.....	38
34	8 LOGICAL-DEVICE class model.....	39
35	8.1 LOGICAL-DEVICE class definition	39
36	8.1.1 LOGICAL-DEVICE class syntax	39
37	8.1.2 LOGICAL-DEVICE class attributes	39
38	8.2 LOGICAL-DEVICE class services	40
39	8.2.1 GetLogicalDeviceDirectory.....	40
40	9 LOGICAL-NODE class model.....	41
41	9.1 LOGICAL-NODE class definition	41
42	9.1.1 LOGICAL-NODE class syntax	41
43	9.1.2 LOGICAL-NODE class attributes	41
44	9.2 LOGICAL-NODE class services	43
45	9.2.1 Overview	43
46	9.2.2 GetLogicalNodeDirectory	

1	9.2.3 GetAllDataValues.....	44
2	10 DATA class model	46
3	10.1 General	46
4	10.2 DATA class definition.....	46
5	10.2.1 DATA class syntax	46
6	10.2.2 DATA class attributes	48
7	10.3 Relation of DATA, common DATA, and compatible DATA classes	55
8	10.4 DATA class services.....	56
9	10.4.1 General definitions and overview.....	56
10	10.4.2 GetDataValue	56
11	10.4.3 SetDataValue.....	57
12	10.4.4 GetDataDirectory	58
13	10.4.5 GetDataDefinition	59
14	11 DATA-SET class model	61
15	11.1 General	61
16	11.2 DATA-SET class definition.....	62
17	11.2.1 DATA-SET class syntax	62
18	11.2.2 DATA-SET class attributes.....	62
19	11.3 DATA-SET class services	63
20	11.3.1 Overview	63
21	11.3.2 GetDataSetValues	63
22	11.3.3 SetDataSetValues.....	64
23	11.3.4 CreateDataSet	65
24	11.3.5 DeleteDataSet	66
25	11.3.6 GetDataSetDirectory	67
26	12 Substitution model	67
27	13 SETTING-GROUP-CONTROL-BLOCK class model.....	70
28	13.1 General	70
29	13.2 SGCB class definition	71
30	13.2.1 SGCB class syntax	71
31	13.2.2 SGCB class attributes.....	72
32	13.3 SGCB class services	73
33	13.3.1 Overview	73
34	13.3.2 SelectActiveSG.....	73
35	13.3.3 SelectEditSG	74
36	13.3.4 SetSGValue.....	75
37	13.3.5 ConfirmEditSGValue	76
38	13.3.6 GetSGValue	76
39	13.3.7 GetSGCBValue.....	77
40	14 REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK class models	79
41	14.1 Overview	79
42	14.2 REPORT-CONTROL-BLOCK class model.....	81
43	14.2.1 Basic concepts	81
44	14.2.2 BUFFERED-REPORT-CONTROL-BLOCK (BRCB) class definition	81
45	14.2.3 BRCB class services.....	91
46	14.2.4 UNBUFFERED-REPORT-CONTROL-BLOCK (URCB) class definition	106
47	14.2.5 URCB class services	107
48	14.3 LOG-CONTROL-BLOCK class model.....	109
49	14.3.1 General	109

1	14.3.2	LCB class definition	110
2	14.3.3	LOG class definition.....	114
3	14.3.4	Procedures to generate the log entries	117
4	14.3.5	LOG services.....	117
5	15	Generic substation event class model (GSE).....	121
6	15.1	Overview	121
7	15.2	GOOSE-CONTROL-BLOCK (GoCB) class	122
8	15.2.1	GoCB definition	122
9	15.2.2	GOOSE service Definitions	124
10	15.2.3	Generic object oriented substation event (GOOSE) message	129
11	15.3	Generic substation state event (GSSE) control block (GsCB)	131
12	15.3.1	GsCB class definition.....	131
13	15.3.2	Generic substation state event (GSSE) control block class attributes.....	133
14	15.3.3	GSSE service definitions.....	134
15	15.3.4	Generic substation state event (GSSE) message	138
16	16	Transmission of sampled value class model.....	140
17	16.1	Overview	140
18	16.2	Transmission of sampled values using multicast	141
19	16.2.1	MSVCB class definition.....	141
20	16.2.2	Multicast sampled value class services	143
21	16.3	Transmission of sampled values using unicast	146
22	16.3.1	USVCB class definition	146
23	16.3.2	Unicast sampled value services	148
24	16.4	Sampled value format.....	151
25	16.4.1	MsvID or UsvID.....	152
26	16.4.2	OptFlds.....	152
27	16.4.3	DatSet	152
28	16.4.4	Sample [1..n]	152
29	16.4.5	SmpCnt	152
30	16.4.6	RefrTm	153
31	16.4.7	ConfRev	153
32	16.4.8	SmpSynch	153
33	16.4.9	SmpRate	153
34	17	CONTROL class model.....	154
35	17.1	Introduction	154
36	17.2	Control with normal security.....	155
37	17.2.1	Direct control with normal security.....	155
38	17.2.2	SBO control with normal security.....	156
39	17.3	Control with enhanced security	157
40	17.3.1	Introduction	157
41	17.3.2	Direct control with enhanced security	157
42	17.3.3	SBO control with enhanced security	158
43	17.4	Time-activated operate	161
44	17.5	CONTROL class service definitions	162
45	17.5.1	Overview	162
46	17.5.2	Service parameter definition.....	163
47	17.5.3	Service specification	164
48	18	Time and time-synchronization model	168
49	18.1	General	168

1	18.2 External information.....	168
2	19 Naming conventions	169
3	19.1 Class naming and class specializations.....	169
4	19.2 Referencing an instance of a class.....	170
5	19.3 Scope.....	172
6	20 File transfer	173
7	20.1 File transfer model.....	173
8	20.1.1 FileName	173
9	20.1.2 FileSize [0..1].....	173
10	20.1.3 LastModified	173
11	20.2 File services	174
12	20.2.1 GetFile	174
13	20.2.2 SetFile	174
14	20.2.3 DeleteFile	175
15	20.2.4 GetFileAttributeValues	176
16	Annex A (normative) ACSI conformance statement.....	177
17	A.1 General	177
18	A.2 ACSI basic conformance statement.....	177
19	A.3 ACSI models conformance statement.....	178
20	A.4 ACSI service conformance statement.....	179
21	Bibliography	183
22	Index	184
23		
24	Figure 1 – Excerpt of conceptual model.....	16
25	Figure 2 – Basic conceptual class model of the ACSI.....	17
26	Figure 3 – Conceptual service model of the ACSI	18
27	Figure 4 – Data and control blocks (conceptionell model)	21
28	Figure 5 – Data (CB) attribute type concept (example).....	22
29	Figure 6 – Overview about GetDirectory and GetDefinition services	31
30	Figure 7 – Access views of a server	33
31	Figure 8 – Normal operation	34
32	Figure 9 – Aborting association	34
33	Figure 10 – Principle of multicast application association.....	38
34	Figure 11 – Class diagram of DATA and DataAttributeType	46
35	Figure 12 – Example of DATA	47
36	Figure 13 – Relation of TrgOp and Reporting.....	53
37	Figure 14 – Relation of DATA classes	55
38	Figure 15 – Excerpt of data class services.....	56
39	Figure 16 – Dynamic creation of data set instances	61
40	Figure 17 – Principles of substitution	68
41	Figure 18 – Basic model of the settings model.....	70
42	Figure 19 – Setting group state machine.....	72
43	Figure 20 – Basic building blocks for reporting and logging	80
44	Figure 21 – BRCB state machine.....	83

1	Figure 23 – Buffer time.....	86
2	Figure 26 – Report example on the use of sequence number	93
3	Figure 29 – Data set members and reporting	96
4	Figure 30 – Report example	98
5	Figure 31 – Log model overview	109
6	Figure 32 – GoCB model.....	121
7	Figure 33 – Specifics for GsCB model	132
8	Figure 34 – Model for transmission of sampled values	140
9	Figure 35 – Principle of the control model.....	154
10	Figure 36 – State machine of direct control with normal security	155
11	Figure 37 – Direct control with normal security.....	155
12	Figure 38 – State machine of SBO control with normal security.....	156
13	Figure 39 – State machine of direct control with enhanced security.....	158
14	Figure 40 – State machine SBO control with enhanced security	159
15	Figure 41 – Select before operate with enhanced security – positive case.....	160
16	Figure 42 – Select before operate with enhanced security – negative case (no status	
17	change)	160
18	Figure 43 – Time-activated operate	162
19	Figure 44 – Time model and time synchronization (principle)	168
20	Figure 45 – Specializations.....	169
21	Figure 46 – Object names and object reference.....	172
22		
23	Table 1 – ACSI classes	20
24	Table 2 – BasicTypes	23
25	Table 3 – ObjectName type	24
26	Table 4 – ObjectReference type	24
27	Table 5 – ServiceError type	25
28	Table 6 – PACKED-LIST type.....	26
29	Table 7 – TimeStamp type.....	26
30	Table 8 – TimeQuality definition	27
31	Table 9 – TimeAccuracy.....	28
32	Table 10 – TriggerConditions type	29
33	Table 11 – SERVER class definition	29
34	Table 12 – TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition	34
35	Table 13 – MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition.....	38
36	Table 14 – LOGICAL-DEVICE (LD) class definition.....	39
37	Table 15 – LOGICAL-NODE (LN) class definition.....	41
38	Table 16 – DATA class definition.....	48
39	Table 17 – DAType definition.....	49
40	Table 18 – Functional constraint values.....	51
41	Table 19 – TrgOp	53
42	Table 20 – COMMON-DATA class definition.....	54
43	Table 21 – DATA-SET (DS) class definition.....	54

1	Table 22 – SGCB class definition	71
2	Table 23 – BRCB class definition	82
3	Table 24 – Report format specification	92
4	Table 25 – ReasonForInclusion	97
5	Table 25 – URCB class definition	106
6	Table 27 – LCB class definition	110
7	Table 28 – LOG class definition.....	115
8	Table 29 – GOOSE control block class definition	122
9	Table 30 – GOOSE message definition.....	130
10	Table 31 – GSSE control block class definition	132
11	Table 32 – GSSE message definition.....	139
12	Table 33 – MSVCB class definition	141
13	Table 34 – USVCB class definition	146
14	Table 35 – Sampled value (SV) format definition	152
15	Table 35 – Control services.....	162
16	Table 37 – Additional cause diagnosis definition	163
17	Table 38 – AddCause semantic	163
18	Table 39 – TimeActivatedOperate response definition	164
19	Table 40 – List of ObjectReferences.....	170
20	Table 41 – FILE class definition.....	173
21	Table A.1 – Basic conformance statement.....	177
22	Table A.2 – ACSI models conformance statement	178
23	Table A.3 – ACSI service conformance statement	179
24	Table A.3 (<i>continued</i>)	180
25	Table A.3 (<i>continued</i>)	181
26	Table A.3 (<i>continued</i>)	182

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 IEC (International Electrotechnical Commission) is a worldwide organisation for standardisation comprising all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote international co-operation on all questions concerning standardisation in the electrical and electronic fields. To this end and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organisations liaising with the IEC also participate in this preparation. The IEC collaborates closely with the International Organisation for Standardisation (ISO) in accordance with conditions determined by agreement between the two organisations.
- 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested National Committees.
- 3) The documents produced have the form of recommendations for international use and are published in the form of standards, technical specifications, technical reports or guides and they are accepted by the National Committees in that sense.
- 4) In order to promote international unification, IEC National Committees undertake to apply IEC International Standards transparently to the maximum extent possible in their national and regional standards. Any divergence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated in the latter.
- 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with one of its standards.
- 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject of patent rights. The 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 system control and associated communications.

This CD of Edition 2 is based on the following documents:

IS	Report on voting
IEC 61850-7-2:2003	57/629/RVD

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

IEC 61850 consists of the following parts, under the general title *Communication networks and systems for utility automation*:

- Part 1: Introduction and overview
- Part 2: Glossary
- Part 3: General requirements
- Part 4: System and project management
- Part 5: Communication requirements for functions and device models
- Part 6: Configuration description language for communication in electrical substations related to IEDs
- Part 7-1: Basic information and communication structure for substation and feeder equipment – Principles and models

- 1 Part 7-2: Basic information and communication structure for substation and feeder equipment
2 – Abstract communication service interface (ACSI)
- 3 Part 7-3: Basic information and communication structure for substation and feeder equipment
4 – Common data classes
- 5 Part 7-4: Basic information and communication structure for substation and feeder equipment
6 – Compatible logical node classes and data classes
- 7 Part 8-1: Specific communication service mapping (SCSM) – Mappings to MMS (ISO/IEC 9506-
8 1 and ISO/IEC 9506-2) and to ISO/IEC 8802-3
- 9 Part 9-1: Specific communication service mapping (SCSM) – Sampled values over serial uni-
10 directional multidrop point to point link
- 11 Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC
12 8802-3
- 13 Part 10: Conformance testing
- 14
- 15 TC57, WG10 is currently preparing the second edition of IEC 61850. During the recent WG
16 meeting, it was decided to circulate in a first step the following parts as CD: Part 6, 7-2, 7-3, 7-
17 4, 8-1 and 9-2. The purpose of these CDs is, to give the national committees a first possibility
18 to comment on the principle direction of Edition 2. There may still be some missing elements in
19 these CDs.
- 20 It is intended, to circulate by the end of 2007 a CDV of Edition 2. For that CDV, other parts, in
21 particular part 1, 5 and 7-1 will be prepared as well.

22

1 INTRODUCTION

2 This document is part of a set of definitions which details a layered utility communication archi-
3 tecture. This architecture has been chosen to provide abstract definitions of classes and ser-
4 vices such that the definitions are independent of specific protocol stacks, implementations,
5 and operating systems.

6 The IEC 61850 series is intended to provide interoperability between a variety of devices.
7 Communication between these devices is achieved by the definition of a hierarchical class
8 model (for example, logical device, logical node, data, data set, report control, or log) and ser-
9 vices provided by these classes (for example, get, set, report, define, delete) in parts IEC
10 61850-7-x.

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

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

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

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

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

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

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

32

33 Editor's Note:

34 This document is based on the experiences made after the publication of the first Edition of
35 this standard. The crucial changes made in this CD are according to the Tissues on part IEC
36 61850-7-2 (Technical Issues) posted at the Tissue database (www.tissue.iec61850.com).

37

38

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

COMMUNICATION NETWORKS AND SYSTEMS FOR 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,
 - publisher/subscriber,
 - 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 for utility automation – Part 2: Glossary*

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

IEC 61850-7-1, *Communication networks and systems for utility automation – Part 7-1: Basic information and communication structure for substation and feeder equipment – Principles and models*

IEC 61850-7-3, *Communication networks and systems for utility automation – Part 7-3: Basic information and communication structure for substation and feeder equipment – Common data classes*

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

1 IEC 61850-8-1: *Communication networks and systems for utility automation – Part 8-1: Specific*
2 *communication service mapping (SCSM) – Mappings to MMS (ISO/IEC 9506-1 and ISO/IEC*
3 *9506-2) and to ISO/IEC 8802-3*

4 3 Terms and definitions

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

7 3.1

8 **class**

9 description of a set of objects that share the same attributes, services, relationships, and se-
10 mantics

11 3.2

12 **client**

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

15 3.3

16 **device**

17 entity that performs control, actuating and/or sensing functions and interfaces to other such en-
18 tities within an automation system

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

20 3.4

21 **external equipment**

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

24 EXAMPLE Transformer, circuit-breaker, line.

25 NOTE 1 Equipment can contain devices.

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

28 3.5

29 **instance (of a class)**

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

32 NOTE Instance is a synonym for the term object.

33 3.6

34 **Logical device**

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

36 3.7

37 **Logical node**

38 entity that represents a typical automation, protection or other function

39 3.8

40 **physical device**

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

42 NOTE Physical devices host logical devices.

1 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
DAType	data attribute type
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
LN	logical node
MC	multicast
MCAA	multicast application association
MMS	manufacturing message specification
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)
SG	setting group

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™	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 General

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) and IEC 61850-7-4 (compatible logical node classes and compatible data classes for utility automation applications) 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 common models (for example, logical node and data classes including their services) are 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.

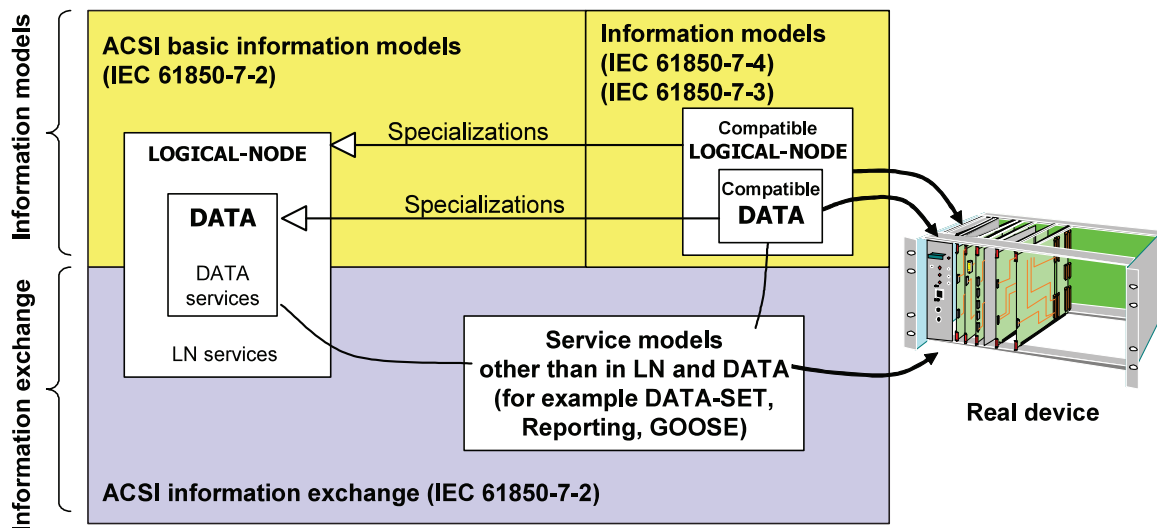


Figure 1 – Excerpt of conceptual model

Other service models required for utility automation systems (for example, data sets and reporting provide specific information exchange services) are also defined in this part of the standard; these models are linked to logical nodes and data. The information exchange services are completely defined in the ACSI. The information models defined in IEC 61850-7-4 reference the services defined in the various models of the ACSI.

5.2 Overview of basic information models

The conceptual models to build the domain-specific information models are:

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

NOTE 1 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)** – contains the information produced and consumed by a group of domain-specific application functions; functions are defined as logical nodes.

- c) **Logical node (LN)** – contains the information produced and consumed by a domain-specific application function, for example, overvoltage protection or circuit-breaker.

- 1 d) **Data** – provide means to define typed information, for example, position of a switch with
2 quality information and timestamp, contained in logical nodes.
- 3 Each of these information models is defined as a class. The classes comprise attributes and
4 services. The conceptual class diagram of the ACSI is depicted in Figure 2.
- 5 NOTE 2 The classes are major building blocks that provide the framework for utility automation device models.
6 Additional details on the modelling and relations between IEC 61850-7-4, IEC 61850-7-3, and this part of IEC 61850
7 can be found in IEC 61850-7-1.

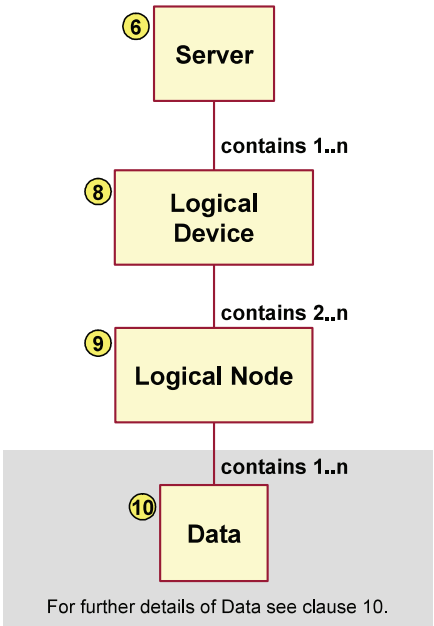


Figure 2 – Basic conceptual class model of the ACSI

- 10 NOTE 3 The numbers in the circles indicate the respective clauses in this part of IEC 61850.
- 11 Each of the following classes has a name and a reference: logical device, logical node, and
12 data.
- 13 EXAMPLE In an implementation the logical device, logical node, data, and data attribute have each an object
14 name (instance name) which is a unique name among classes of the same container to which they belong. In addi-
15 tion, each of the four has an ObjectReference (path name) which is a concatenation of all object names from each
16 container. The four object names (one per column) can be concatenated.

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

- 17
- 18 **5.3 Overview of the other service models**
- 19 In addition to the models listed above, the ACSI comprises the following models that provide
20 services operating on data, data attributes, and data sets.
- 21 a) **Data Set** – permits the grouping of data and data attributes. Used for direct access and
22 for reporting and logging.
- 23 b) **Substitution** – supports replacement of a process value by another value.
- 24 c) **Setting group control** – defines how to switch from one set of setting values to another
25 one and how to edit setting groups.



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

- 1 NOTE 2 The class diagrams are conceptual. Details are defined in the respective clauses. Comprehensive dia-
2 grams are contained in IEC 61850-7-1. The data class may be defined recursively. The operations for substitution
3 and control are restricted to the lowest level in the **data** class. The DataAttributes may be defined recursively as
4 well.
- 5 The logical node is one of the major building blocks that have associations to most of the other
6 information exchange models, for example, report control, log control, and setting control.
- 7 Any other information exchange service model, for example, report control, log control, and
8 setting control shall inherit the ObjectName and ObjectReference as depicted in Figure 2.
- 9 NOTE 3 The class models and services are defined using an object-oriented approach allowing for the mapping of
10 class models and services to different application layer and middle ware solutions.

1 **5.4 Overview of ACSI services**

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

3 **Table 1 – ACSI classes**

<u>SERVER model (Clause 6)</u> GetServerDirectory <u>ASSOCIATION model (Clause 7)</u> Associate Abort Release <u>LOGICAL-DEVICE model (Clause 8)</u> GetLogicalDeviceDirectory <u>LOGICAL-NODE model (Clause 9)</u> GetLogicalNodeDirectory GetAllDataValues <u>DATA model (Clause 10)</u> GetDataValue SetDataValue GetDataDirectory GetDataDefinition <u>DATA-SET model (Clause 11)</u> GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory <u>Substitution model (Clause 12)</u> SetDataValue GetDataValue <u>SETTING-GROUP-CONTROL-BLOCK model (Clause 13)</u> SelectActiveSG SelectEditSG SetSGValues ConfirmEditSGValues GetSGValues GetSGCBValues <u>REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK model (Clause 14)</u> BUFFERED-REPORT-CONTROL-BLOCK: Report GetBRCBValues SetBRCBValues UNBUFFERED-REPORT-CONTROL-BLOCK: Report GetURCBValues SetURCBValues	<u>LOG-CONTROL-BLOCK model</u>⁴₅ GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues <u>Generic substation event model – GSE (Clause 15)</u> GOOSE SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues GSSE SendGSSEMessage GetGsReference GetGSSEDataOffset GetGsCBValues SetGsCBValues <u>Transmission of sampled values model (Clause 16)</u> MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendMSVMessage GetMSVCBValues SetMSVCBValues UNICAST-SAMPLE-VALUE-CONTROL-BLOCK: SendUSVMessage GetUSVCBValues SetUSVCBValues <u>Control model (Clause 17)</u> Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate <u>Time and time synchronization (Clause 18)</u> TimeSynchronization <u>FILE transfer model (Clause 20)</u> GetFile SetFile DeleteFile GetFileAttributeValues
--	---

1 5.5 Type definitions

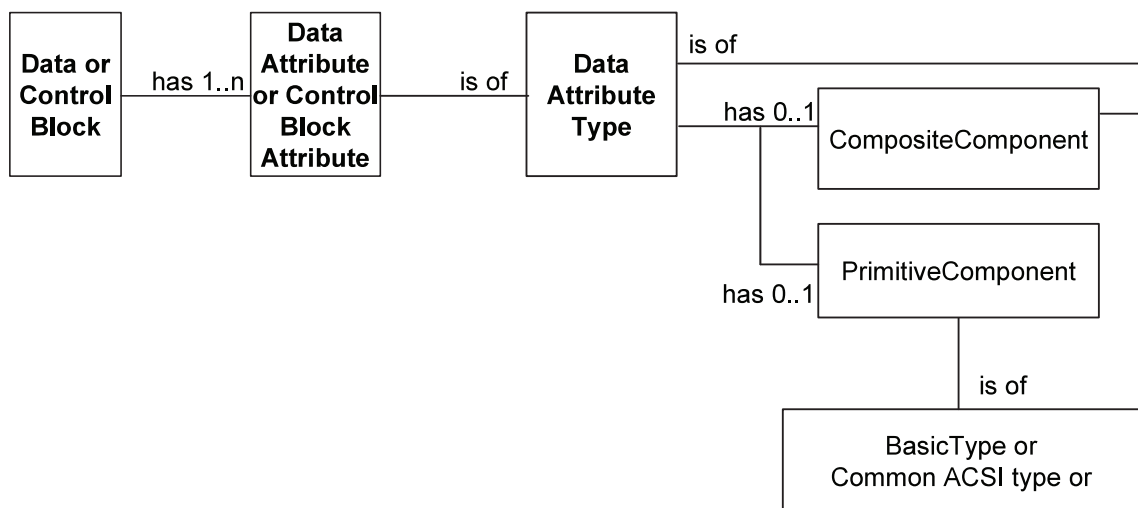
2 5.5.1 General

3 The crucial modelling aspect is the structure of data objects (e.g. a measurement or a position
4 of a breaker). Data objects usually comprise several data attributes like the current value, time
5 stamp, quality, and configuration attributes. This part of IEC 61850 defines the basic formal
6 model of the information models for application data and for the information exchange. The
7 definitions throughout this part and parts IEC 61850-7-3 and IEC 61850-7-4 are based on types
8 defined in this clause.

9 Editor's Note: This clause 5.5.1 General may be moved to the Ed2 of IEC 61850-7-1. IEC
10 61850-7-2 may focus on the formal definitions only!

11 IEC 61850-7-2 and IEC 61850-7-3 shall use the types that are defined in the following sub-
12 clauses in order to define the specific data for the information models in IEC 61850-7-4 and the
13 control block models in this part of IEC 61850 (for example, report control blocks).

14 The basic concept of the data (control blocks) and their attributes are depicted in Figure 4.



16 **Figure 4 – Data and control blocks (conceptionell model)**

17 The data (e.g., measured value) shown on the left side has one or more data attributes (current
18 value, time stamp, quality information). Each data attribute is of a specific type (the data attrib-
19 ute type). The data attribute type can be primitive or can comprise structured components. The
20 primitive component can be of a BasicType (see 5.5.2) or a common ACSI type (see 5.5.3),
21 The composite component can, e.g., be a vector comprising a magnitude and a angle value,
22 each composed of an integer and a floating value,

23 NOTE 1 The modelling method for data objects is also used for the definition of control blocks that define the
24 the information exchange models, e.g. reporting and logging.

25 Figure 5 shows an example of a data (and control block) object.

26 NOTE 2 The data used in the example is artificial (it is not a data object defined in IEC 61850-7-4 and IEC
27 61850-7-3).

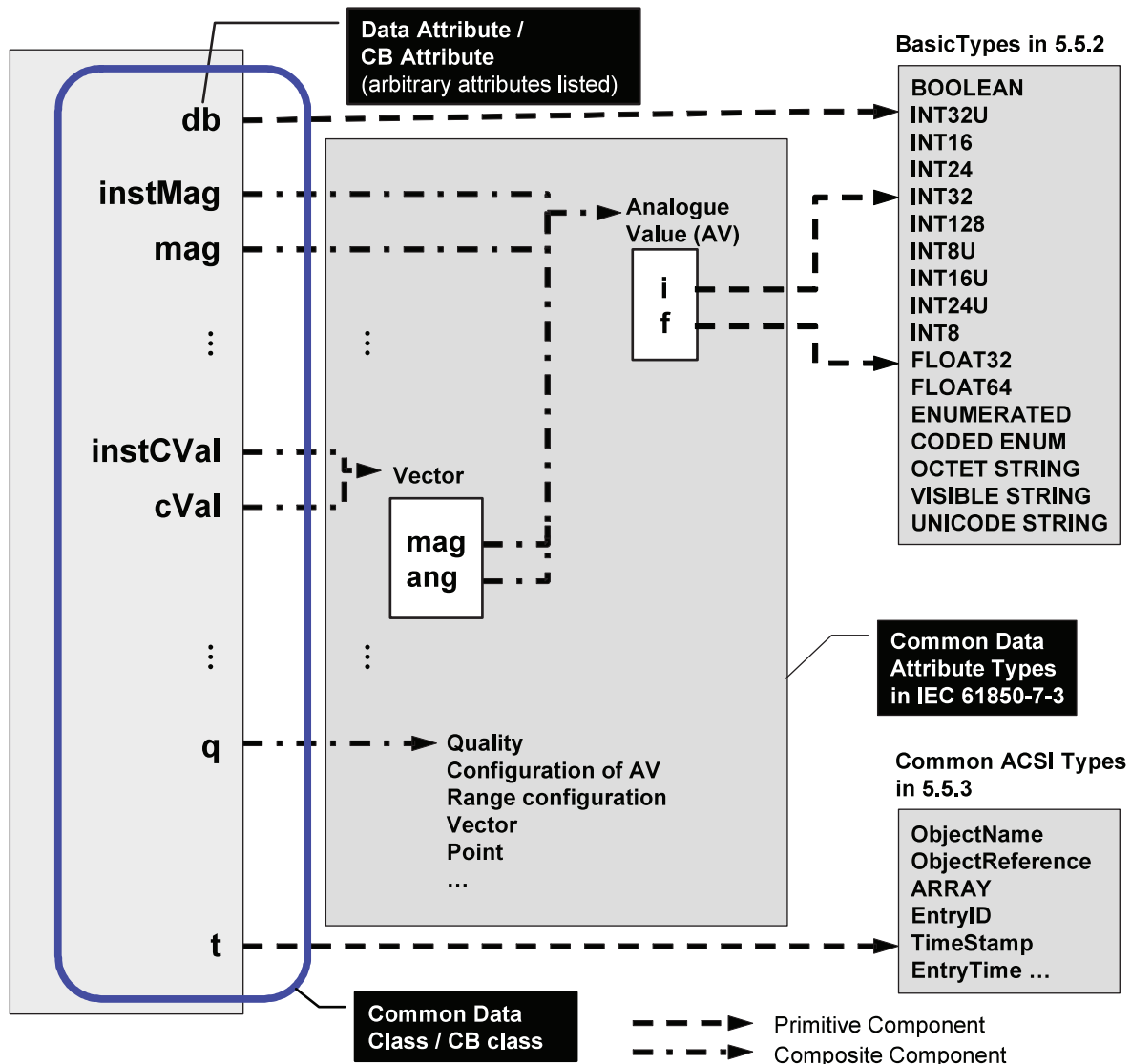


Figure 5 – Data (CB) attribute type concept (example)

The data (respectively a control block, CB) are defined as a list of data attributes. Data attributes are named and structured. The attribute names are shown on the right in the left box (db, instMag, ...). Each of the attributes has a well defined structure. The structure is defined in the corresponding data attribute type. There are two kinds of data attribute types: the primitive components and the composite components.

Primitive components (depicted as dashed line) are typed with the types defined in clause 5.5.2 (basic types) or clause 5.5.3 (common ACSI types). Composite components (depicted as dotted and dashed lines) are typed with types defined, e.g. in IEC 61850-7-3 for utility automation applications and in IEC 61400-25-2 for wind power plant applications.

Useful lists of data attributes (components) have been defined for re-use in various application domains. These lists are called common data classes (control block classes). Common data classes are defined in IEC 61850-7-3 for utility automation applications and in IEC 61400-25-2 for wind power plant applications. Further application domains may require additional common data classes.

1 5.5.2 Basic types

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

3 Table 2 – BasicTypes

BasicTypes			
Name	Value range	Remark	Used by
BOOLEAN	0,1		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
INT24	-8 388 608 to 8 388 607	for TimeStamp type	IEC 61850-7-2
INT32	-2 147 483 648 to 2 147 483 647		IEC 61850-7-3 IEC 61850-7-2
INT128	-2**127 to (2**127)-1	Required for counters	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		IEC 61850-7-2
INT32U	Unsigned integer, 0 to 4 294 967 295		IEC 61850-7-3 IEC 61850-7-2
CtxInt	Value range and mapping depend on the use and SCSM		IEC 61850-7-3
FLOAT32	Range of values and precision as specified by IEEE 754 single- precision floating point		IEC 61850-7-3
FLOAT64	Range of values and precision as specified by IEEE 754 double- 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 al- lowed	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 encod- ing in a SCSM	IEC 61850-7-3 IEC 61850-7-2
OCTET STRING	Max. length shall be defined where type is used ^a		IEC 61850-7-3 IEC 61850-7-2
VISIBLE STRING	Max. length shall be defined where type is used ^a		IEC 61850-7-3 IEC 61850-7-2
UNICODE STRING	Max. length shall be defined where type is used ^a		IEC 61850-7-3

^a The length suffix shall have the format "...STRINGnn" where "nn" is the length in characters.

5.5.3 Common ACSI types

5.5.3.1 General

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

5.5.3.2 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
NOTE Clause 19 specifies constraints on the use of the type ObjectName.			

5.5.3.3 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.	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 "[.]" shall indicate an option. The inner square bracket "[. ...]" shall indicate further names of recursively nested definitions.

NOTE 1 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.

NOTE 2 Clause 19 specifies constraints on the use of the type ObjectReference.

1 5.5.3.4 PHYCOMADDR type

2 The type **PHYCOMADDR** shall represent a physical communication address (e.g. media access address, priority, and other information) as defined by a SCSM.

4 5.5.3.5 ARRAY type

5 The type **ARRAY** shall be as defined as follows:

6 **ARRAY [0..m] OF p**

7 with $m \geq 0$

8 $p =$ Common data attribute type or

9 BasicType or

10 Common ACSI type (except ARRAY type)

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

13 The ARRAY type shall be applied for DataAttributes only.

14 NOTE Common data attribute types for utility automation applications are defined in IEC 61850-7-3.

15 5.5.3.6 ServiceError type

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

18 **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	IEC 61850-7-2

19
20 Additional **ServiceError** values for negative service responses (originated in the application,
21 for example, additional cause diagnosis for control-related services) shall be as specified in the
22 appropriate service models.

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

24 5.5.3.7 EntryID type

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

27 NOTE 1 The **EntryID** (handle) allows a client to re-synchronize, for example, with the sequence of the events
28 stored in the IED. The syntax and semantic of the **EntryID** are outside the scope of this standard.

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

5.5.3.8 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 SCSSM. No access to individual members of the list is required	IEC 61850-7-3 IEC 61850-7-2

5.5.3.9 TimeStamp type

5.5.3.9.1 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 18.

NOTE 1 The TimeStamp type relies on requirements specified in Clause 18. The reader should first read that clause. The presentation of the TimeStamp is defined in the SCSSMs.

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

5.5.3.9.2 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 $b_i \cdot 2^{22-i}$; Order = b0, b1, b2, b3, ...	M
TimeQuality	TimeQuality		M

5.5.3.9.3 TimeStamp attributes

5.5.3.9.3.1 SecondSinceEpoch

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

NOTE SecondSinceEpoch corresponds with the Unix epoch.

5.5.3.9.3.2 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 $l = 0$ to 23 of $b_l \cdot 2^{22-l}$ 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 $1/2^{**1}$ (= 0,5 s) if only the first bit is used; or may be $1/2^{**2}$ (= 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.

5.5.3.9.3.3 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		O
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 takes into account 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.

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

ClockNotSynchronized: The attribute **clockNotSynchronized** shall indicate that the time source of the sending device is not synchronized with the external UTC time.

TimeAccuracy: The attribute **TimeAccuracy** shall represent the time accuracy class of the time source of the sending device relative to the external UTC time. 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 1 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)

5.5.3.10 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** classes in IEC 61850-7-4. The **EntryTime** type is used for all IEC 61850-7-2 class definitions. 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.

5.5.3.11 TriggerConditions type

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

NOTE 1 The TriggerConditions are used in this part of IEC 61850 and in IEC 61850-7-3.

Table 10 – TriggerConditions type

TriggerConditions type			
Attribute name	Attribute type	TriggerOption (TrgOp) for use in DataAttributes	Value/value range/explanation
	PACKED LIST		
data-change	BOOLEAN	dchg	Trigger used in DATA-Attributes determined by common DATA classes of IEC 61850-7-3
quality-change	BOOLEAN	qchg	Trigger used in DATA-Attributes determined by common DATA classes of IEC 61850-7-3
data-update	BOOLEAN	dupd	Trigger used in DATA-Attributes determined by common DATA classes of IEC 61850-7-3
integrity	BOOLEAN	–	Trigger whose value (time) can be set by a service or by configuration; independent of an instance of DATA
general-interrogation	BOOLEAN	–	Trigger whose value (initiate general interrogation) can be set by a service or by configuration; independent of an instance of DATA

The TriggerOption (**TrgOp**) shall be used in the definition of **DataAttributes** to indicate on which change/update the value of an instance of a **DataAttribute** may be reported or logged.

NOTE 2 Details on the use of **TriggerConditions** are defined in 10.2.2.4.3 and Clause 14.

6 SERVER class model

6.1 SERVER class definition

6.1.1 SERVER class syntax

The class **SERVER** shall represent the externally visible behaviour of a device. The **SERVER** shall be a composition as defined in Table 11.

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

Table 11 – SERVER class definition

SERVER class		
Attribute name	Attribute type	Value/value range/explanation
ServiceAccessPoint [1..n]	(*)	(*) Type is SCSM specific
LogicalDevice [1..n]	LOGICAL-DEVICE	
File [0..n]	FILE	
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.

1 6.1.2 SERVER class attributes

2 6.1.2.1 ServiceAccessPoint [1..n]

3 The attribute **ServiceAccessPoint** shall identify a **SERVER** within the scope of a system.

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

8 6.1.2.2 LogicalDevice [1..n]

9 The attribute **LogicalDevice** shall identify a **LogicalDevice** that is contained in a **SERVER**.

10 6.1.2.3 File [0..n]

11 The attribute **File** shall identify a **File** contained in a **SERVER**.

12 6.1.2.4 TPAAppAssociation [0..n] – two-party application association

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

15 NOTE Details can be found in Clause 7.

16 6.1.2.5 MCAAppAssociation [0..n] – multicast application association

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

19 NOTE Details can be found in Clause 7.

20 6.2 Server class services

21 6.2.1 Overview of directory and GetDefinition services

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

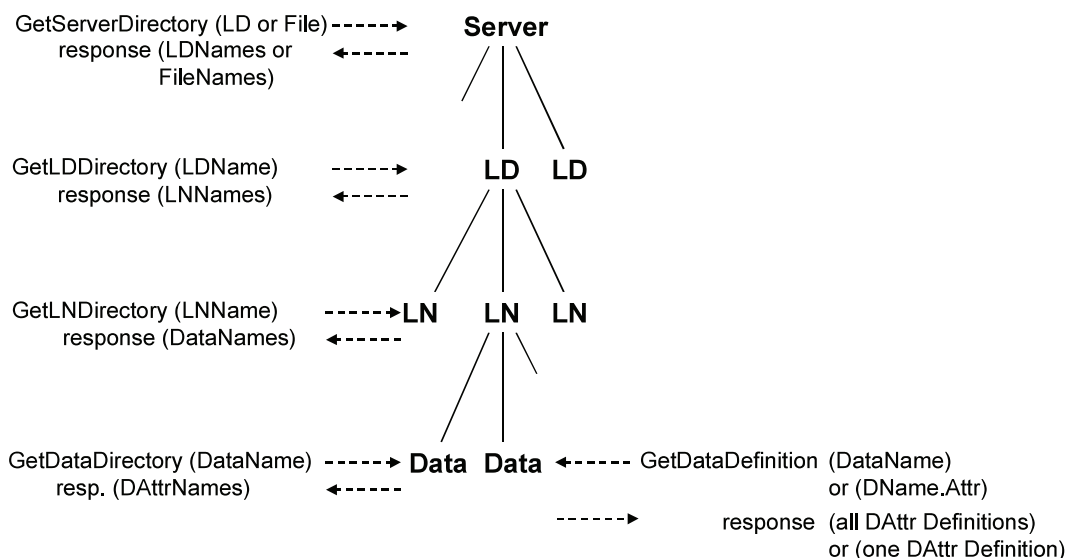


Figure 6 – Overview about GetDirectory and GetDefinition services

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.

6.2.2 GetServerDirectory

6.2.2.1 GetServerDirectory parameter table

A client shall use the **GetServerDirectory** service to retrieve a list of the names of all **LOGICAL-DEVICES** or **Files** made visible and thus accessible to the requesting client by the addressed **SERVER**.

NOTE The visible instances are those that are defined within a given view (see Clause 7 for details on the view concept).

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

6.2.2.2 Request

6.2.2.2.1 ObjectClass

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

- **LOGICAL-DEVICE**
- **FILE**

6.2.2.3 Response+

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

6.2.2.3.1 Reference [0..n]

The parameter **Reference** shall contain the ObjectReference of the **LOGICAL-DEVICE** or the **FileName**.

NOTE The **FileName** type is VISIBLE STRING255.

6.2.2.4 Response–

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

7 Application association model

7.1 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 requirements are implemented by the SCSMs.

IEC 62351 shall be used for the definition of the appropriate access control roles (e.g Role Based Access Control) when available.

Editor's note: IEC 62351-6 TS Ed.1 "Data and communication security - Part 6: Security for IEC 61850 profiles" is about to be published as a TS. Part 7-2 will include in the CDV a section on Role Based Access Control and Authentication/Identity Establishment. Role based accessibility to the objects will be dealt offline and configured via the SCL. Allocation of role to users can be changed online (add, remove, modify). Therefore new services are also required to the new users model.

7.2 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).

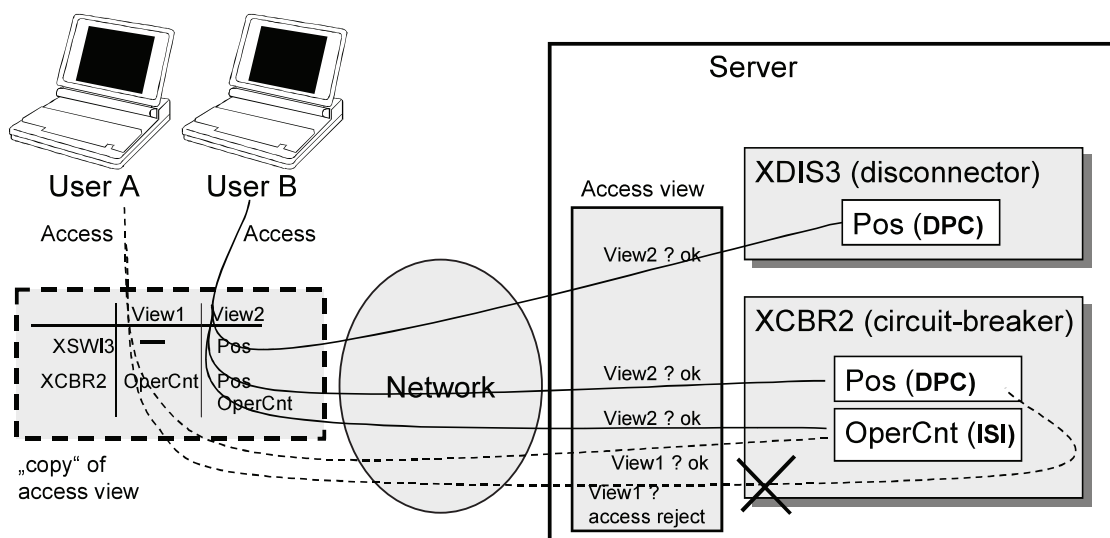
Application associations provide a mechanism for controlling the access to the instances of a device (access control).

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.

7.3 Access control

The access control model provides the capability to restrict the access of a specific client to class instances, class instance attributes, and ACSI services acting upon class instances of a specific server. The ACSI server contains a set of, for example, **LOGICAL-DEVICES**, **LOGICAL-NODEs**, **DATA**, or report controls. The set of instances visible (and therefore accessible) to a client is restricted on the basis of the identification of the client and the access control definition of the server. This restricted set is called a virtual access view. A virtual access view may not only restrict the visibility of instances or attributes but also the supported service. The concept of a virtual access view is illustrated in Figure 7.

NOTE 1 The virtual access view is the authentication's view of the IED's data model.



IEC 401/03

Figure 7 – Access views of a server

Two users (A and B) have different virtual access views (view1 and view2) of the server. View 1 allows just one **DATA** (**XCBR.OperCnt**) to be accessed remotely. View 2 allows all **DATA** to be accessed.

The intention of IEC 61850 is to implement the **virtual access view in the server** of a device, thus providing **access restriction to any user** who tries to access the instances. Independent of the implementation in the device, additional access restriction may be implemented at the user side, for example, local password or simply a key on the keyboard.

If a view hides a mandatory instance of an attribute of a **DATA** then this hidden attribute shall be implemented as required by the **DATA**.

NOTE 2 A view restricts the visibility to some users only.

A client (or a subscriber in the case of multicast application association) shall be identified by authentication parameters passed to the server when establishing the association with the server (two-party application association) or when sending information over multicast application associations.

NOTE 3 Mechanisms at the client side are outside the scope of this standard. A user may also use a "copy" of the access view to restrict the access at the client side.

NOTE 4 The details of access control including structure and content of authentication parameter are defined in the SCSMs.

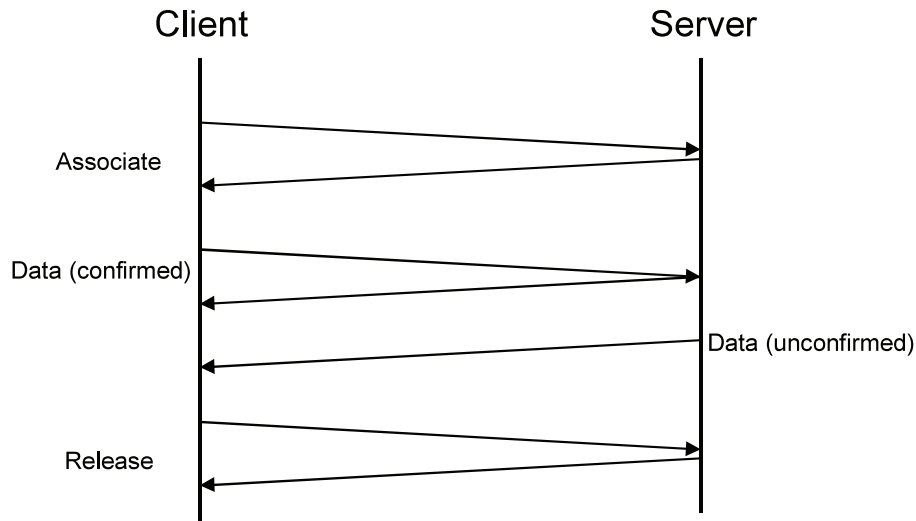
7.4 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model

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

7.4.1.1 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.

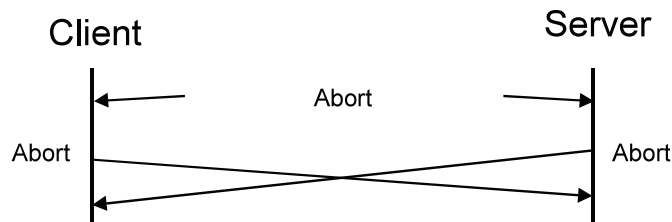
- 1 The services for associate, data exchange, and association release of the two-party application
2 association class is depicted in Figure 8.



IEC 402/03

Figure 8 – Normal operation

- 5 The abort service for the two-party application association class is depicted in Figure 9.



IEC 403/03

Figure 9 – Aborting association

- 8 The **TWO-PARTY-APPLICATION-ASSOCIATION** (TPAA) class shall be defined as in Table 12.

Table 12 – 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")		

1 7.4.1.2 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class attributes

2 7.4.1.2.1 AssociationId

3 The attribute **AssociationId** shall define the identification used to identify the application as-
4 sociations.

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

7 7.4.1.2.2 AuthenticationParameter

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

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

12 7.4.2 Two-party application association services

13 7.4.2.1 Overview

14 For **TWO-PARTY-APPLICATION-ASSOCIATION** the following services are defined.

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

15

16 7.4.2.2 Associate

17 7.4.2.2.1 Associate parameter

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

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

20

21 7.4.2.2.2 Request

22 7.4.2.2.2.1 ServerAccessPointReference

23 The parameter **ServerAccessPointReference** shall identify the server, with which the applica-
24 tion association shall be established.

1 7.4.2.2.2 AuthenticationParameter

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

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

7 7.4.2.2.3 Response+

8 AssociationId

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

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

11 7.4.2.2.4 Result

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

14 7.4.2.2.5 Response–

15 The parameter **Response–** shall indicate that the service request failed. The appropriate **Ser-**
16 **viceError** shall be returned.

17 7.4.2.3 Abort

18 7.4.2.3.1 Abort parameter

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

Parameter name
Request
AssociationId
Reason
Indication
AssociationId
Reason

22

23 7.4.2.3.2 Request

24 7.4.2.3.2.1 AssociationId

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

28 7.4.2.3.2.2 Reason

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

1 **7.4.2.3.3 Indication**2 **7.4.2.3.3.1 AssociationId**

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

4 **7.4.2.3.3.2 Reason**

5 The parameter **Reason** shall define the reason for abrupt termination the application associa-
6 tion.

7 **7.4.2.4 Release**8 **7.4.2.4.1 Release parameter**

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

Parameter name
Request
AssociationId
Response+
AssociationId
Result
Response–
ServiceError

12

13 **7.4.2.4.2 Request**14 **7.4.2.4.2.1 AssociationId**

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

16 **7.4.2.4.3 Response+**17 **7.4.2.4.3.1 AssociationId**

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

19 **7.4.2.4.3.2 Result**

20 The parameter **Result** shall indicate, if the termination of the application association was suc-
21 cessful.

22 **7.4.2.4.4 Response–**

23 The parameter **Response–** shall indicate that the service request failed. The appropriate Ser-
24 viceError shall be returned.

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

7.5 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class

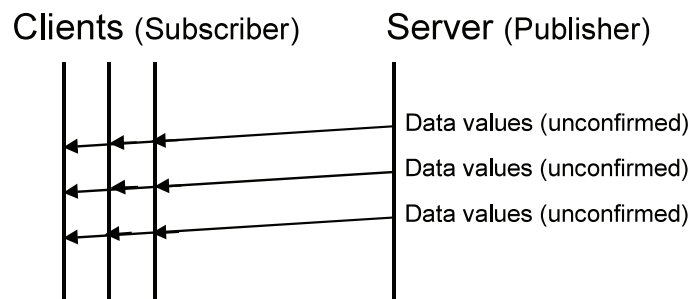
7.5.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition

7.5.1.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class syntax

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 10.



IEC 404/03

Figure 10 – Principle of multicast application association

The **MULTICAST-APPLICATION-ASSOCIATION (MCAA)** shall be as defined in Table 13.

Table 13 – 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")		

7.5.1.2 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class attributes

7.5.1.2.1 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.

8 LOGICAL-DEVICE class model

8.1 LOGICAL-DEVICE class definition

8.1.1 LOGICAL-DEVICE class syntax

The **LOGICAL-DEVICE** (**LD**) shall be a composition of **LOGICAL-NODE** as defined in Table 14.

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

Table 14 – LOGICAL-DEVICE (LD) class definition

LOGICAL-DEVICE class		
Attribute name	Attribute type	Value/value range/explanation
LDName	ObjectName	Instance name of an instance of LOGICAL-DEVICE
LDRef	ObjectReference	Path-name of an instance of LOGICAL-DEVICE
LogicalNode [2..n]	LOGICAL-NODE	IEC 61850-7-4 specifies specialized classes of LOGICAL-NODE
Services GetLogicalDeviceDirectory		

8.1.2 LOGICAL-DEVICE class attributes

8.1.2.1 LDName – logical device name

The attribute **LDName** shall unambiguously identify a **LOGICAL-DEVICE** within the scope of a system.

8.1.2.2 LDRef – logical device ObjectReference

The attribute **LDRef** shall be the unique path-name of a **LOGICAL-DEVICE**:

LDName

NOTE The **LOGICAL-DEVICE** is the root of tree. Therefore the **LDName** and **LDRef** are identical. For conceptual reasons they are both included in the table.

8.1.2.3 LogicalNode [2..n]

The attribute **LogicalNode** shall identify a **LOGICAL-NODE** that is contained in a **LOGICAL-DEVICE**.

Each **LOGICAL-DEVICE** shall have one and only one **LOGICAL-NODE-ZERO** (**LLN0**), one and only one **LOGICAL-NODE-PHYSICAL-DEVICE** (**LPHD**), and it may have one or more other **LOGICAL-NODEs**.

NOTE The **LLN0**, **LPHD**, and other logical nodes are defined in IEC 61850-7-4 for utility automation applications.

8.2 LOGICAL-DEVICE class services

8.2.1 GetLogicalDeviceDirectory

8.2.1.1 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**.

NOTE The visible instances are those that are defined within a given view (see Clause 7 for details on the view concept).

Parameter Name
Request
LDReference
Response+
Reference [2..n]
Response–
ServiceError

8.2.1.2 Request

8.2.1.2.1 LDReference – logical device ObjectReference

The parameter **LDReference** shall contain the ObjectReference **LDRef** of a **LOGICAL-DEVICE**.

8.2.1.3 Response+

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

8.2.1.3.1 LNReference [3..n] – logical node ObjectReference

The parameter **LNReference** shall contain the ObjectReference **LNRef** of a **LOGICAL-NODE** from the referenced **LOGICAL-DEVICE**.

8.2.1.4 Response–

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

9 LOGICAL-NODE class model

9.1 LOGICAL-NODE class definition

9.1.1 LOGICAL-NODE class syntax

The **LOGICAL-NODE** shall be a composition of **DATA**, **DATA-SET**, **BRCB**, **URCB**, **LCB**, **LOG**, **SGCB**, **GoCB**, **GsCB**, **MSVCB**, and **USVCB** as defined in Table 15.

Table 15 – LOGICAL-NODE (LN) class definition

LOGICAL-NODE class		
Attribute name	Attribute type	Explanation
LNName	ObjectName	Instance name of an instance of LOGICAL-NODE
LNRef	ObjectReference	Path-name of an instance of LOGICAL-NODE
Data [1..n]	DATA	
DataSet [0..n]	DATA-SET	
BufferedReportControlBlock [0..n]	BRCB	
UnbufferedReportControlBlock [0..n]	URCB	
Log [0..n]	LOG	
LogControlBlock [0..n]	LCB	
IF compatible LN class defined in IEC 61850-7-4 equals LLNO		
SettingGroupControlBlock [0..1]	SGCB	
GOOSEControlBlock [0..n]	GoCB	
GSSEControlBlock [0..n]	GsCB	
MulticastSampledValueControlBlock [0..n]	MSVCB	
UnicastSampledValueControlBlock [0..n]	USVCB	
Services GetLogicalNodeDirectory GetAllDataValues		
NOTE 1 IEC 61850-7-4 defines specialized logical node classes – the compatible logical node classes, for example, XCBR representing circuit-breakers.		

The definition of **LOGICAL-NODEs** for the utility automation domain is refined by the definition of specific **DATA** 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 2 IEC 61850-7-4 defines further attributes for **LOGICAL-NODEs**; for example, the mode (behaviour: ON, BLOCKED, TEST, 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** (named **Mod**).

9.1.2 LOGICAL-NODE class attributes

9.1.2.1 LNName – Logical node name

The attribute **LNName** shall unambiguously identify **LOGICAL-NODE** within the scope of **LOGICAL-DEVICE**.

1 **9.1.2.2 LNRef – Logical node ObjectReference**

2 The attribute **LNRef** shall be the unique path-name of a **LOGICAL-NODE**.

3 The ObjectReference LNRef shall be:

LDName/LNName

4

5 **9.1.2.3 Data [1..n]**

6 The attribute **Data** shall identify **DATA** (see Clause 10) that is contained in the **LOGICAL-**
7 **NODE**.

8 NOTE IEC 61850-7-4 defines standardized **DATA** called compatible **DATA** classes.

9 **9.1.2.4 DataSet [0..n]**

10 The attribute **DataSet** shall identify a **DATA-SET** (see Clause 11) that is contained in the
11 **LOGICAL-NODE**.

12 **9.1.2.5 BufferedReportControlBlock [0..n]**

13 The attribute **BufferedReportControlBlock** shall identify a **BRCB** (see 14.2) that is contained
14 in the **LOGICAL-NODE**.

15 **9.1.2.6 UnbufferedReportControlBlock [0..n]**

16 The attribute **UnbufferedReportControlBlock** shall identify an **URCB** (see 14.2) that is con-
17 tained in the **LOGICAL-NODE**.

18 **9.1.2.7 Log [0..n]**

19 The attribute **Log** shall identify the **LOG** (see 14.3.3) that is contained in the **LOGICAL-NODE**.

20 **9.1.2.8 LogControlBlock [0..n]**

21 The attribute **LogControlBlock** shall identify a **LCB** (see 14.3) that is contained in the
22 **LOGICAL-NODE**.

23 **9.1.2.9 SettingGroupControlBlock [0..1]**

24 The attribute **SettingGroupControl** shall identify the **SGCB** (see Clause 13) that is contained
25 in a **LLNO**.

26 **9.1.2.10 GOOSEControlBlock [0..n]**

27 The attribute **GOOSEControlBlock** shall identify a **GoCB** (see 15.2) that is contained in the
28 **LLNO**.

29 **9.1.2.11 GSSEControlBlock [0..n]**

30 The attribute **GSSEControlBlock** shall identify the **GsCB** (see 15.3) that is contained in the
31 **LLNO**.

1 9.1.2.12 MulticastSampledValueControlBlock [0..n]

2 The attribute **MulticastSampledValueControlBlock** shall identify a **MSVCB** (see 16.2) that is
3 contained the **LLNO**.

4 9.1.2.13 UnicastSampledValueControlBlock [0..n]

5 The attribute **UnicastSampledValueControlBlock** shall identify a **USVCB** (see 16.3) that is
6 contained in the **LLNO**.

7 9.2 LOGICAL-NODE class services

8 9.2.1 Overview

9 For **LOGICAL-NODE** 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 contained in the LOGICAL-NODE

10

11 9.2.2 GetLogicalNodeDirectory

12 9.2.2.1 GetLogicalNodeDirectory parameter table

13 A client shall use the **GetLogicalNodeDirectory** service to retrieve a list of the ObjectRefer-
14 ences of all instances of a requested class made visible and thus accessible to the requesting
15 client by the referenced **LOGICAL-NODE**.

16 NOTE The visible instances are those that are defined within a given view (see Clause 7 for details on
17 the view concept).

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

18

19 9.2.2.2 Request

20 9.2.2.2.1 LNReference

21 The parameter **LNReference** shall contain the ObjectReference **LNRef** of the **LOGICAL-**
22 **NODE**.

23 9.2.2.2.2 ACSIClass

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

26 The client shall select one identification for one of the following ACSI class models:

27 **DATA, DATA-SET, BRCB, URCB, LCB, LOG, SGCB, GoCB, GsCB, MSVCB** and **USVCB**.

1 9.2.2.3 Response+

2 The parameter **Response+** shall indicate that the service request succeeded. A successful re-
3 sult shall return the following parameter.

4 InstanceName [0..n]

5 If the ACSI class model requested equals DATA-SET, BRCB, URCB, LCB, LOG, SGCB, GoCB,
6 GsCB, MSVCB, or USVCB:

7 The parameter **InstanceName** shall contain an ObjectName of the requested class.

8 If the ACSI class model requested equals DATA:

9 The parameter **InstanceName** shall contain an ObjectName (in case of one level of DATA) or
10 a concatenation of ObjectNames separated by "." (in case of several levels of DATA)."

11 In the case where the referenced **LOGICAL-NODE** does not contain the requested ACSI class,
12 the server shall indicate that no ACSI class model exists in this **LOGICAL-NODE**.

13 9.2.2.4 Response–

14 The parameter **Response–** shall indicate that the service request failed. The appropriate Ser-
15 viceError shall be returned.

16 9.2.3 GetAllDataValues

17 9.2.3.1 GetAllDataValues parameter table

18 A client shall use the **GetAllDataValues** service to retrieve all **DataAttribute** values (having
19 the same **FunctionalConstraint**) of all **DATA** made visible and thus accessible to the re-
20 questing client by the referenced **LOGICAL-NODE**.

21 NOTE The visible instances are those that are defined within a given view (see Clause 7 for details on the
22 view concept).

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

23

24 9.2.3.2 Request

25 9.2.3.2.1 LNReference

26 The parameter **LNReference** shall contain the ObjectReference **LNRef** of the **LOGICAL-**
27 **NODE**.

1 **9.2.3.2.2 FunctionalConstraint [0..1]**

2 The parameter **FunctionalConstraint** shall contain the functional constraint parameter (**FC**) to
3 filter the respective **DataAttributes** of all **DATA** contained in the **LOGICAL-NODE**. The **FC**
4 shall be as defined in 10.2.2.4.2.

5 **9.2.3.3 Response+**

6 The parameter **Response+** shall indicate that the service request succeeded. A successful re-
7 sult shall return the following parameters.

8 **9.2.3.3.1 DataAttributeReference [1..n]**

9 The parameter **DataAttributeReference** shall contain the ObjectReference of a
10 **DataAttribute** contained in the **LOGICAL-NODE** that shall be returned according to the
11 value of the **FunctionalConstraint** received in the request.

12 NOTE The ObjectReference **DataAttributeReference** is defined in 10.2.2.4.

13 **9.2.3.3.2 DataAttributeValue [1..n]**

14 The parameter **DataAttributeValue** shall contain the value of a **DataAttribute** of the **DATA**
15 contained in the referenced **LOGICAL-NODE**. Only values of those **DataAttributes** that have
16 the functional constrained equal to the value of the parameter **FunctionalConstraint** in the
17 service request shall be returned.

18 **9.2.3.4 Response–**

19 The parameter **Response–** shall indicate that the service request failed. The appropriate Ser-
20 viceError shall be returned.

10 DATA class model

10.1 General

DATA classes represent meaningful information of applications located in an automation devices. The values of **DATA** instances can, for example, be written (**SetDataValue**) and read (**GetDataValue**). IEC 61850-7-4 specifies a list of common and utility-domain-specific – simple and complex – **DATA**, for example, **Pos** for position, **OilFil** for oil filtration. The composition of **DATA** in IEC 61850-7-4 is based on common templates (the common **DATA** classes, **CDC**) specified in IEC 61850-7-3. The concept of **DATA** classes is introduced in this clause. Any set of **DATA** (or parts of **DATA**) instances may be grouped to build **DATA-SET** instances applying the **CreateDataSet** service. **DATA-SET** instances can, for example, be written (**SetDataSetValues**) or read (**GetDataSetValues**).

NOTE 1 The consequences of setting values to instances of **DATA** is outside this part of IEC 61850. IEC 61850-7-3 and IEC 61850-7-4 define many utility-domain-specific **DATA** classes. These definitions provide information on the actions to be taken by the receiving application, for example, changing the **DATA Mode** from ON to TEST changes the state of the respective instance to test mode behaviour as defined in IEC 61850-7-4.

NOTE 2 The client queries values of **DATA** (**DATA-SET**) from a server using the service **GetDataValue** (**GetDataSetValues**). Services for unsolicited/spontaneous transmission of values of **DATA** from a server to clients (sometimes known as information report, traps, or spontaneous transmission) require a careful design. Uncontrolled spontaneous transmission may congest the network. Services for a controlled reporting are specified in Clause 14.

10.2 DATA class definition

10.2.1 DATA class syntax

The **DATA** class is a key element in IEC 61850. The class diagram in Figure 11 is intended as an introduction to the formal **DATA** class definition.

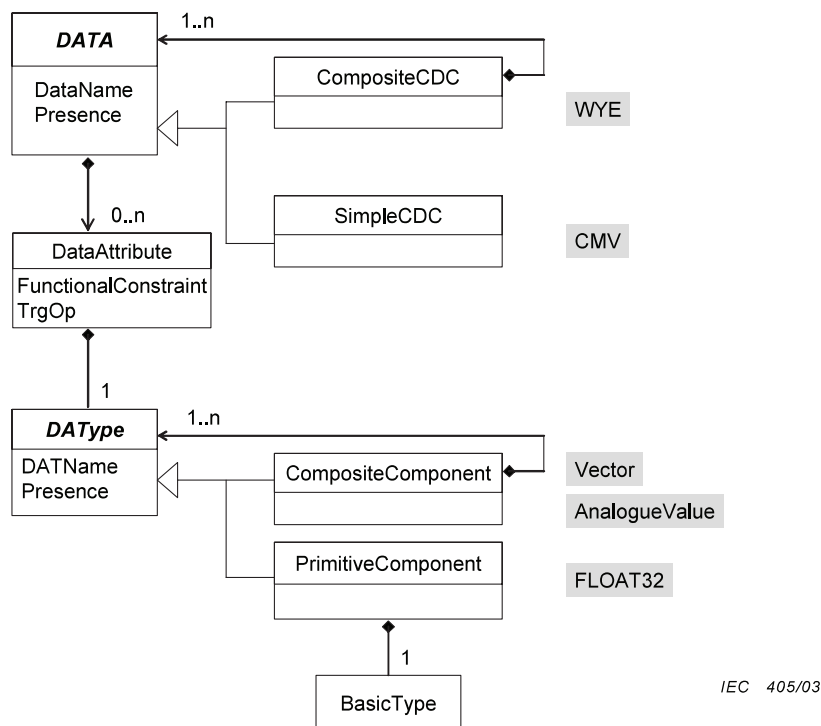


Figure 11 – Class diagram of **DATA** and **DataAttributeType**

NOTE 1 The example in Figure 11 uses definitions (for example, **WYE**, **CMV**, **Vector**, and **AnalogueValue** common **DATA** classes) found in IEC 61850-7-3. A comprehensive introduction to the modelling of **DATA** can be found in IEC 61850-7-1.

1 The **DATA** is a class that has a **DataName**, an indication (Presence) if the **DATA** is mandatory
2 (present) or optional (not-present), and **DataAttributes**.

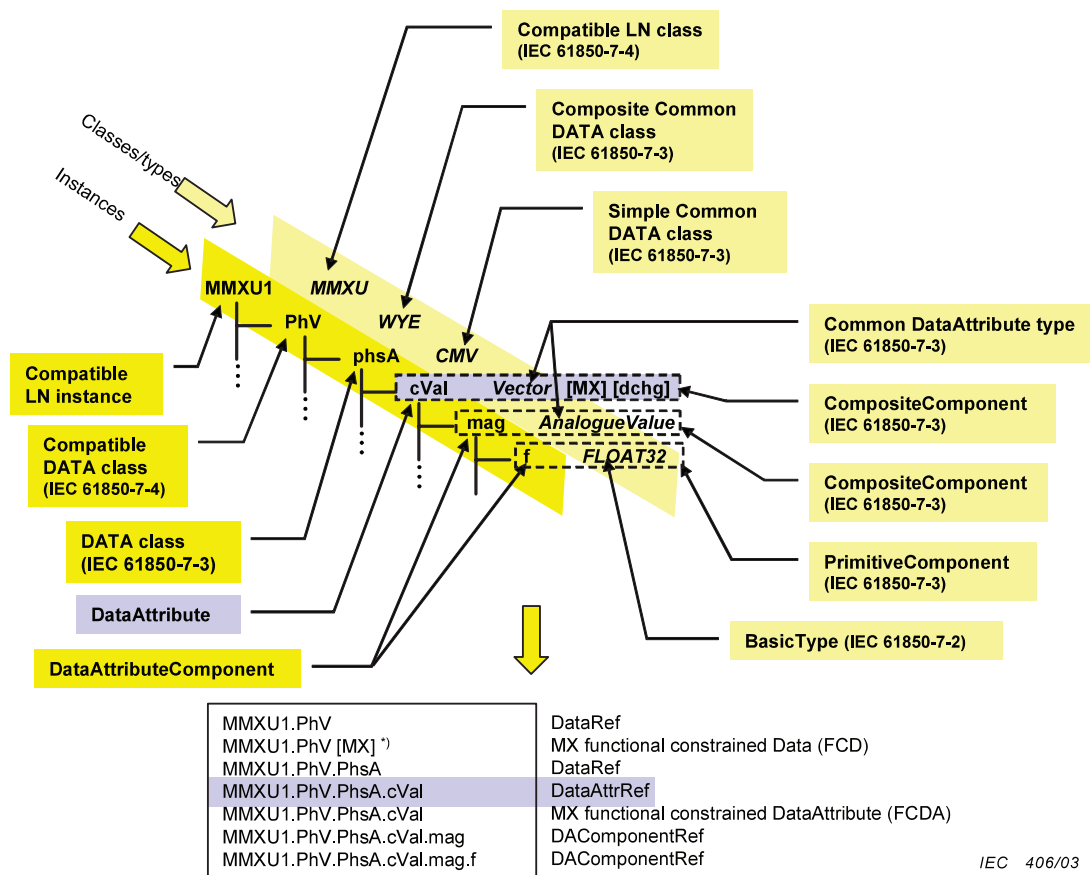
3 NOTE 2 The **DATA** class is an abstract class that is an auxiliary means to construct the primitive and composite
4 common data classes.

5 NOTE 3 The following examples used in the text refer to Figure 12.

6 The **DataAttributes** (for example, **cVal** – complex value) are used to build a **SimpleCDC** (simple
7 common data class) and **CompositeCDC** (composite common data class). **SimpleCDC**
8 shall have a **DataName**, a **Presence**, and **DataAttributes** (for example, **DataName** = **phsA**,
9 **Presence** = **Mandatory**, and **DataAttribute** = **cVal**). The **CompositeCDC** is constructed by one
10 or more **SimpleCDC** and/or **DataAttributes** (for example, **CDC WYE** comprising a **SimpleCDC**
11 **CMV**, etc.).

12 The **DAType** has already been introduced in 5.5.1.

13 Figure 12 depicts an excerpt of a **DATA** instance (contained in a **LOGICAL-NODE** **MMXU1**).
14 The instance of the **LOGICAL-NODE** with the name **MMXU1** (instantiated from **MMXU**) is
15 composed of the instance of the **DATA** phase voltage named **PhV** (instantiated from **WYE**),
16 which is composed of phase A voltage **phsA** (instantiated from **CMV**), which is composed of
17 complex value **cVal** (of type **Vector**), which is composed of voltage **mag** (of type
18 **AnalogueValue**), which is composed of floating-point value **f** (of type **FLOAT32**). The **DataAt**-
19 **tribute** has additionally the functional constraint **FC** = **MX** (measurand) and the trigger option
20 **TrgOp** = **dchg** (data-change).



*) NOTE The notation [MX] is used for explanation purposes only. SCSMs may use a different notation.

Figure 12 – Example of DATA

NOTE 4 The explanation of the **DATA** class refers to the example shown in Figure 12. The example uses some definitions from IEC 61850-7-3 just to demonstrate the formal definition of the **DATA** class. A complete definition of the compatible classes is defined in IEC 61850-7-3.

The references for the various levels are listed at the bottom of the figure.

The **DATA** shall have the structure defined in Table 16.

The inheritance and relations between the classes **DATA**, **CompositeCDC**, **SimpleCDC**, and **DAType** shall be as shown in Figure 11.

The table notation does not easily show the inheritance. Therefore the class diagram in Figure 11 shall be normative. The tables and the class diagrams shall be used together.

Table 16 – DATA class definition

DATA class		
Attribute name	Attribute type	Value/value range/explanation
DataName	ObjectName	Instance name of an instance of DATA, for example, PhV (1st level), phsA (2nd level)
DataRef	ObjectReference	Path-name of an instance of DATA, for example, MMXU1.PhV or for example, MMXU1.PhV.PhsA
Presence	BOOLEAN	Indicates mandatory/optional
DataAttribute [0..n] DataAttributeType FunctionalConstraint TrgOp [0..1]	DAType FC TriggerConditions	For example, Vector class of IEC 61850-7-3 for example, MX for example, dchg
Specializations of DATA		
CompositeCDC [0..n]	DATA	For example, WYE class of IEC 61850-7-3
SimpleCDC [0..n]	COMMON-DATA	For example, CMV class of IEC 61850-7-3
Services GetDataValue SetDataValue GetDataDirectory GetDataDefinition		

An instance of a **DATA** class may contain zero or more instances of a **CompositeCDC**, **SimpleCDC** or a **DataAttribute**. However, they cannot all be absent, so at least one of these elements shall be present.

NOTE 5 The structure of a **DATA class** is recursive since a **CompositeCDC** is also of type **DATA class**. The level of recursion may be restricted by a SCSM, so the number of levels of recursion of **CompositeCDCs** is normally no greater than 1.

NOTE 6 **DATA** or part of a **DATA** may be referenced in a **DATA-SET**. The persistent existence of **DATA** is expected as long as they are referenced as members of a **DATA-SET**. A system has to take special measures to ensure their existence.

10.2.2 DATA class attributes

10.2.2.1 DataName

The attribute **DataName** shall unambiguously identify a **DATA** within the scope of a **LOGICAL-NODE**.

10.2.2.2 DataRef – data ObjectReference

The attribute **DataRef** shall be the unique path-name of a **DATA**.

1 The ObjectReference **DataRef** shall be:

LDName/LNName.DataName[.DataName[. ...]]

2 NOTE Nesting depends on the concrete definition of a **DATA** class.

3 10.2.2.3 Presence

4 The attribute **Presence** of type **BOOLEAN** shall define if a **DATA** within a **compositeCDC** or a
5 **LOGICAL-NODE** is mandatory (**Presence** = **TRUE**) or optional (**Presence** = **FALSE**).

6 NOTE The Presence attribute is a conceptional attribute only. The common data classes as defined, e.g., in IEC
7 61850-7-3 use a column "M/O/C" (mandatory / optional / conditional) to represent if a data shall be present in an
8 implementation.

9 10.2.2.4 DataAttribute

10 10.2.2.4.1 DataAttributeType

11 10.2.2.4.1.1 General

12 The attribute **DataAttributeType** of type **DAType** shall define a data attribute.

13 10.2.2.4.1.2 DAType syntax

14 The **DAType** shall be as defined in Table 17.

15 Table 17 – DAType definition

DAType		
Attribute name	Attribute type	Value/value range/explanation
DATName	ObjectName	Instance name of an instance of DAType , for example, cVal (1st level), mag (2nd level), f (3rd level)
DATRef	ObjectReference	Path-name of an instance of DAType for example MyLD/MMXU1.PhV.phsA.cVal for example MyLD/MMXU1.PhV.phsA.cVal.mag or for example MyLD/MMXU1.PhV.phsA.cVal.mag.f
Presence	BOOLEAN	Indicates mandatory/optional
Specializations of DAType		
CompositeComponent [0..n]	DAType	For example, mag in Vector class of IEC 61850-7-3 for example, f in AnalogueValue of IEC 61850-7-3
PrimitiveComponent [0..1]	BasicType	For example, FLOAT32 class of IEC 61850-7-3 for f
NOTE 1 An instance of a DAType may contain 0 or more instances of a CompositeComponent or a Primitive-DAT . However, they cannot both be absent, so at least one of these elements must be present.		
NOTE 2 The structure of a DAType is recursive since a CompositeComponent is also of type DAType . The level of recursion may be restricted by a SCSM, so the number of levels of recursion of CompositeComponents is normally no greater than 2.		

16

17 DATName – data attribute type name

18 The attribute **DATName** shall unambiguously identify a **DAType** within the scope of a **DataAt-**
19 **tribute** or a nested **DataAttribute**.