



57/891/CD

COMMITTEE DRAFT (CD)

IEC/TC or SC: TC 57	Project number IEC 61850-7-2 Ed.2	
Title of TC/SC: Power systems management and associated information exchange	Date of circulation 2007-05-25	Closing date for comments 2007-09-07
	Also of interest to the following committees	
Supersedes document 57/890/MCR		
Functions concerned: <input type="checkbox"/> Safety <input type="checkbox"/> EMC <input type="checkbox"/> Environment <input type="checkbox"/> Quality assurance		
Secretary: A. Huber, Germany	THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES. RECIPIENTS OF THIS DOCUMENT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.	

Title:

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

(Titre) :

Introductory note

Comments and changes as discussed during the editors' meeting in Zug, 2007-07

- Markups in text
- Move all non-mapping specific definitions from 8-1 to 7-2, e.g. SCL Control Block model

Karlheinz Schwarz

Copyright © 2007 International Electrotechnical Commission, IEC. All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

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.....	16
5.1 General.....	16
5.2 Overview of basic information models.....	16
5.3 Overview of the other service models.....	17
5.4 Overview of ACSI services.....	20
5.5 Type definitions	21
5.5.1 General	21
5.5.2 Basic types	23
5.5.3 Common ACSI types.....	24
6 SERVER class model	29
6.1 SERVER class definition.....	29
6.1.1 SERVER class syntax.....	29
6.1.2 SERVER class attributes.....	30
6.2 Server class services.....	30
6.2.1 Overview of directory and GetDefinition services	30
6.2.2 GetServerDirectory	31
7 Application association model	32
7.1 Introduction	32
7.2 Concept of application associations	32
7.3 Access control.....	32
7.4 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model.....	33
7.4.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition	33
7.4.2 Two-party application association services	35
7.5 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class	38
7.5.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition.....	38
8 LOGICAL-DEVICE class model.....	39
8.1 LOGICAL-DEVICE class definition	39
8.1.1 LOGICAL-DEVICE class syntax	39
8.1.2 LOGICAL-DEVICE class attributes.....	39
8.2 LOGICAL-DEVICE class services	40
8.2.1 GetLogicalDeviceDirectory.....	40
9 LOGICAL-NODE class model.....	41
9.1 LOGICAL-NODE class definition.....	41
9.1.1 LOGICAL-NODE class syntax	41
9.1.2 LOGICAL-NODE class attributes.....	41
9.2 LOGICAL-NODE class services	43
9.2.1 Overview	43
9.2.2 GetLogicalNodeDirectory	43

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.....	62

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

1 INTERNATIONAL ELECTROTECHNICAL COMMISSION

2
3
4 **COMMUNICATION NETWORKS AND SYSTEMS FOR POWER UTILITY**
5 **AUTOMATION –**

6
7 **Part 7-2: Basic information and communication structure –**
8 **Abstract communication service interface (ACSI)**
9

10
11 **FOREWORD**

- 12 1) The IEC (International Electrotechnical Commission) is a worldwide organisation for standardisation comprising
13 all national electrotechnical committees (IEC National Committees). The object of the IEC is to promote interna-
14 tional co-operation on all questions concerning standardisation in the electrical and electronic fields. To this end
15 and in addition to other activities, the IEC publishes International Standards. Their preparation is entrusted to
16 technical committees; any IEC National Committee interested in the subject dealt with may participate in this
17 preparatory work. International, governmental and non-governmental organisations liaising with the IEC also
18 participate in this preparation. The IEC collaborates closely with the International Organisation for Standardisa-
19 tion (ISO) in accordance with conditions determined by agreement between the two organisations.
- 20 2) The formal decisions or agreements of the IEC on technical matters express, as nearly as possible, an interna-
21 tional consensus of opinion on the relevant subjects since each technical committee has representation from all
22 interested National Committees.
- 23 3) The documents produced have the form of recommendations for international use and are published in the form
24 of standards, technical specifications, technical reports or guides and they are accepted by the National Com-
25 mittees in that sense.
- 26 4) In order to promote international unification, IEC National Committees undertake to apply IEC International
27 Standards transparently to the maximum extent possible in their national and regional standards. Any diver-
28 gence between the IEC Standard and the corresponding national or regional standard shall be clearly indicated
29 in the latter.
- 30 5) The IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any
31 equipment declared to be in conformity with one of its standards.
- 32 6) Attention is drawn to the possibility that some of the elements of this International Standard may be the subject
33 of patent rights. The IEC shall not be held responsible for identifying any or all such patent rights.

34 International Standard IEC 61850-7-2 has been prepared by IEC technical committee 57:
35 Power system control and associated communications.

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

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

37
38 This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

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

- 41 Part 1: Introduction and overview
- 42 Part 2: Glossary
- 43 Part 3: General requirements
- 44 Part 4: System and project management
- 45 Part 5: Communication requirements for functions and device models
- 46 Part 6: Configuration description language for communication in electrical substations re-
47 lated to IEDs
- 48 Part 7-1: Basic information and communication structure for substation and feeder equipment
49 – 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 technologies 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 generateon,
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

1

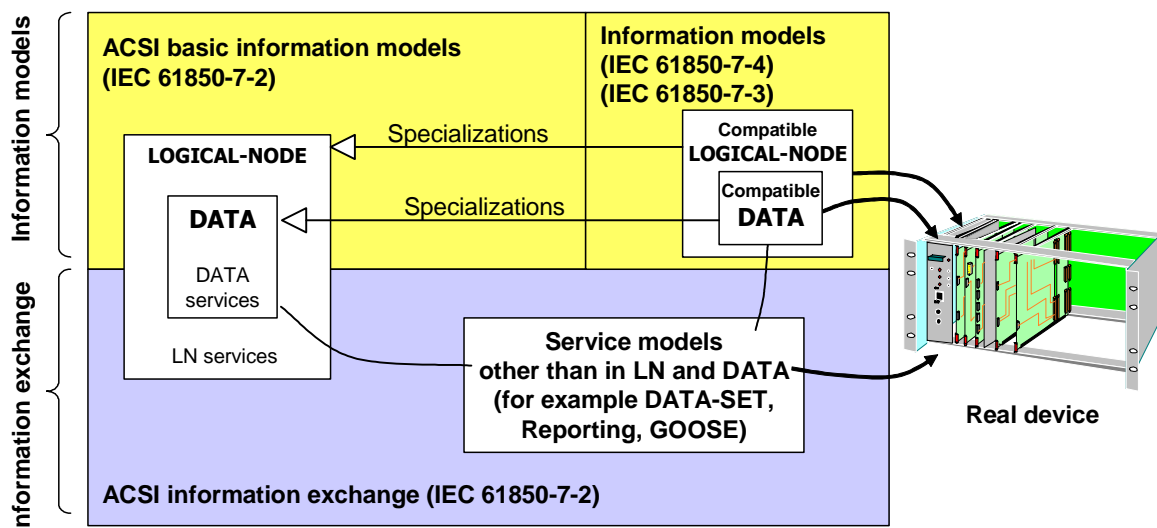
1 5 ACSI overview and basic concepts

2 5.1 General

3 The models of the ACSI provide

- 4 – the definition of the basic model of the utility information models contained in
- 5 IEC 61850-7-3 (common data classes for utility automation applications) and IEC
- 6 61850-7-4 (compatible logical node classes and compatible data classes for utility auto-
- 7 mation applications) and
- 8 – the definition of information exchange service models.

9 The information models and information exchange services are interwoven. From a descriptive
10 point of view, the two aspects are separated to some degree (see the excerpt shown in Fig-
11 ure 1). The common models (for example, logical node and data classes including their ser-
12 vices) are applied in IEC 61850-7-3 and IEC 61850-7-4 to define many specialized information
13 models for utility automation models, or in IEC 61400-25-2 to define specialized models for
14 wind power plant applications.



15

16

Figure 1 – Excerpt of conceptual model

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

22 5.2 Overview of basic information models

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

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

26 NOTE 1 A server has two roles: to communicate with a client (most service models in IEC 61850 provide
27 communication with client devices) and to send information to peer devices (for example, for sampled values).

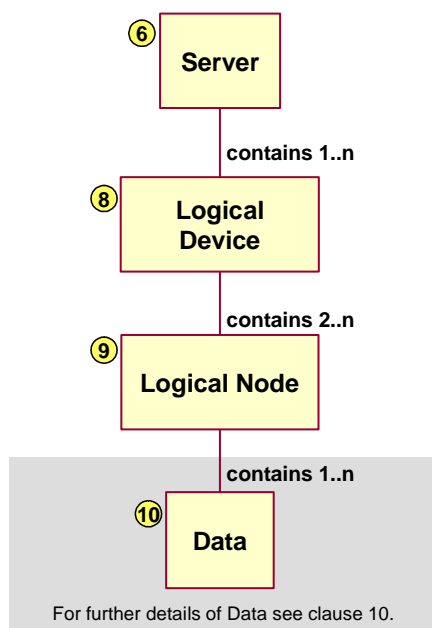
- 28 b) **Logical device (LD)** – contains the information produced and consumed by a group of
29 domain-specific application functions; functions are defined as logical nodes.

- 30 c) **Logical node (LN)** – contains the information produced and consumed by a domain-
31 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.



8

9

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 addition,
 15 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.

- 1 d) **Report control and logging** – describe the conditions for generating reports and logs
- 2 based on parameters set by the client. Reports may be triggered by changes of process
- 3 data values (for example, state change or dead band) or by quality changes. Logs can be
- 4 queried for later retrieval. Reports may be sent immediately or deferred. Reports provide
- 5 change-of-state and sequence-of-events information exchange.

- 6 e) **Control blocks for generic substation event (GSE)** – supports a fast and reliable sys-
- 7 tem-wide distribution of input and output data values; peer-to-peer exchange of IED binary
- 8 status information, for example, a trip signal.

- 9 f) **Control blocks for transmission of sampled values** – fast and cyclic transfer of sam-
- 10 ples, for example, of instrument transformers.

- 11 g) **Control** – describes the services to control, for example, devices.

- 12 h) **Time and time synchronization** – provides the time base for the device and system.

- 13 i) **File transfer** – defines the exchange of large data blocks such as programs
- 14 An overview of the conceptual service model of the ACSI is shown in Figure 3.

This class shall only be available if support is explicitly stated in the definition of a compatible LN class

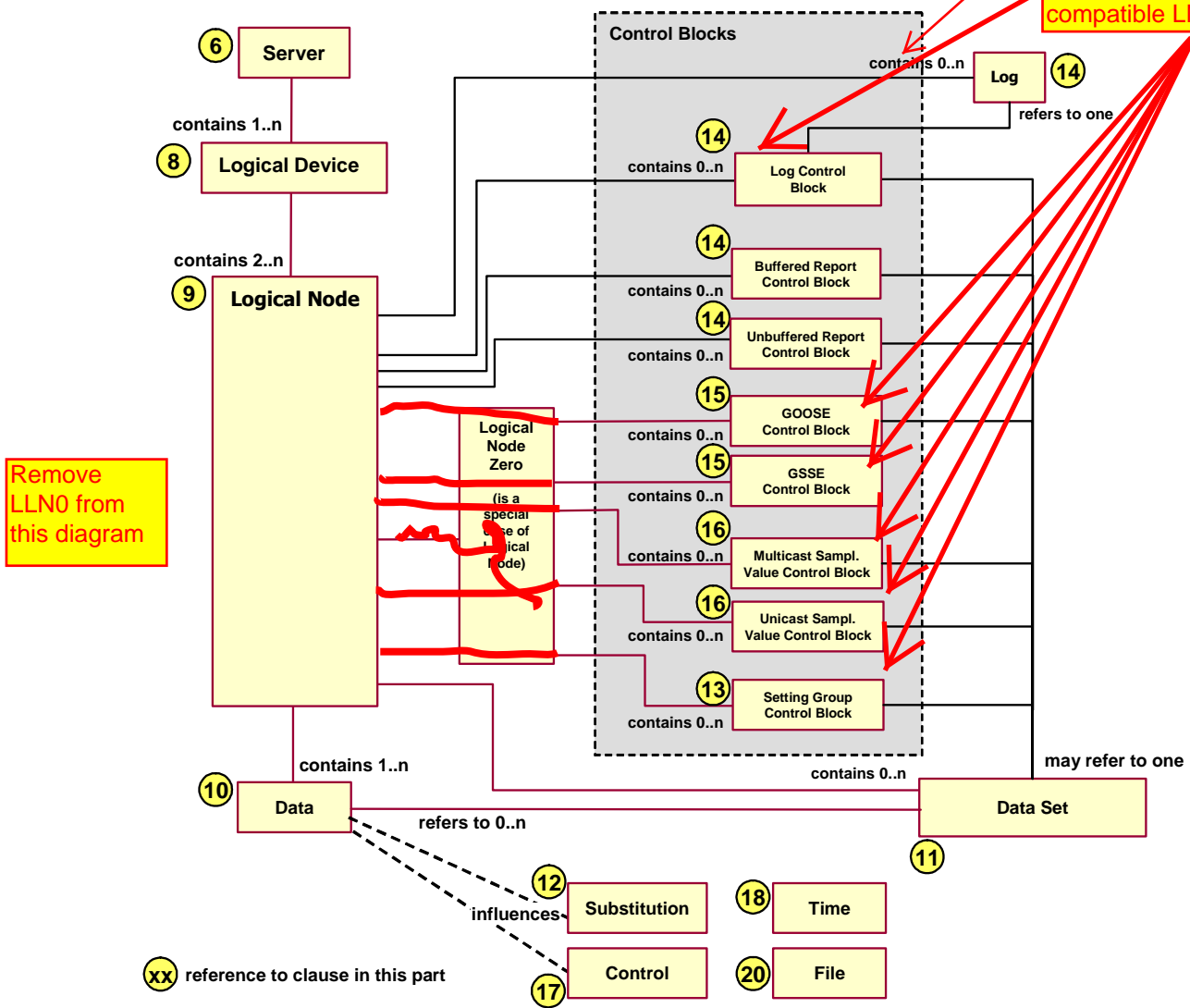


Figure 3 – Conceptual service model of the ACSI

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

- 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**

<p><u>SERVER model (Clause 6)</u> GetServerDirectory</p> <p><u>ASSOCIATION model (Clause 7)</u> Associate Abort Release</p> <p><u>LOGICAL-DEVICE model (Clause 8)</u> GetLogicalDeviceDirectory</p> <p><u>LOGICAL-NODE model (Clause 9)</u> GetLogicalNodeDirectory GetAllDataValues</p> <p><u>DATA model (Clause 10)</u> GetDataValue SetDataValue GetDataDirectory GetDataDefinition</p> <p><u>DATA-SET model (Clause 11)</u> GetDataSetValues SetDataSetValues CreateDataSet DeleteDataSet GetDataSetDirectory</p> <p><u>Substitution model (Clause 12)</u> SetDataValue GetDataValue</p> <p><u>SETTING-GROUP-CONTROL-BLOCK model (Clause 13)</u> SelectActiveSG SelectEditSG SetSGValues ConfirmEditSGValues GetSGValues GetSGCBValues</p> <p><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</p>	<p><u>LOG-CONTROL-BLOCK model</u>⁴₅ GetLCBValues SetLCBValues QueryLogByTime QueryLogAfter GetLogStatusValues</p> <p><u>Generic substation event model – GSE (Clause 15)</u> GOOSE SendGOOSEMessage GetGoReference GetGOOSEElementNumber GetGoCBValues SetGoCBValues GSSE SendGSSEMessage GetGsReference GetGSSEDataOffset GetGsCBValues SetGsCBValues</p> <p><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</p> <p><u>Control model (Clause 17)</u> Select SelectWithValue Cancel Operate CommandTermination TimeActivatedOperate</p> <p><u>Time and time synchronization (Clause 18)</u> TimeSynchronization</p> <p><u>FILE transfer model (Clause 20)</u> GetFile SetFile DeleteFile GetFileAttributeValues</p>
--	---

1 5.5 Type definitions

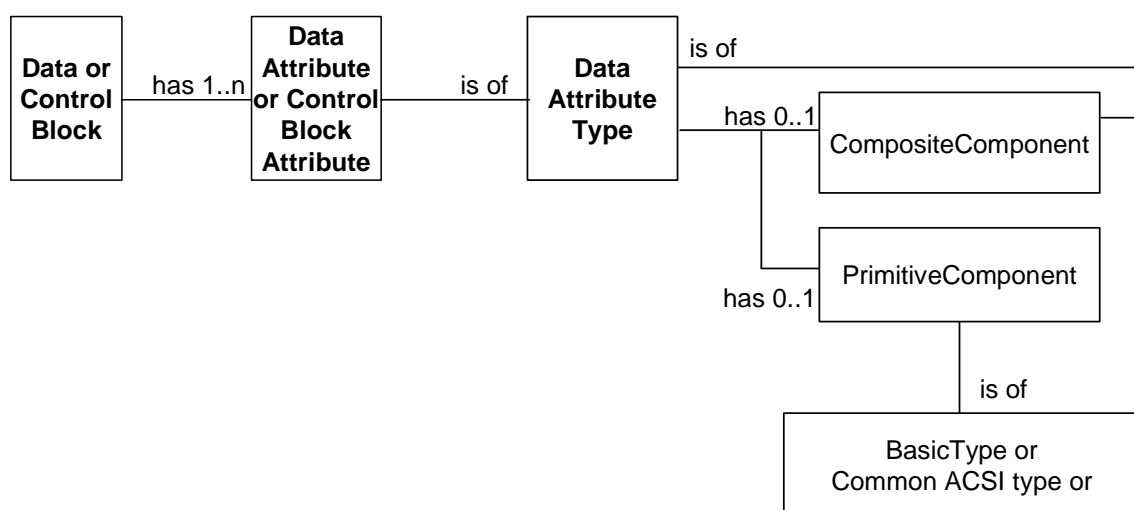
2 5.5.1 General

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.



15

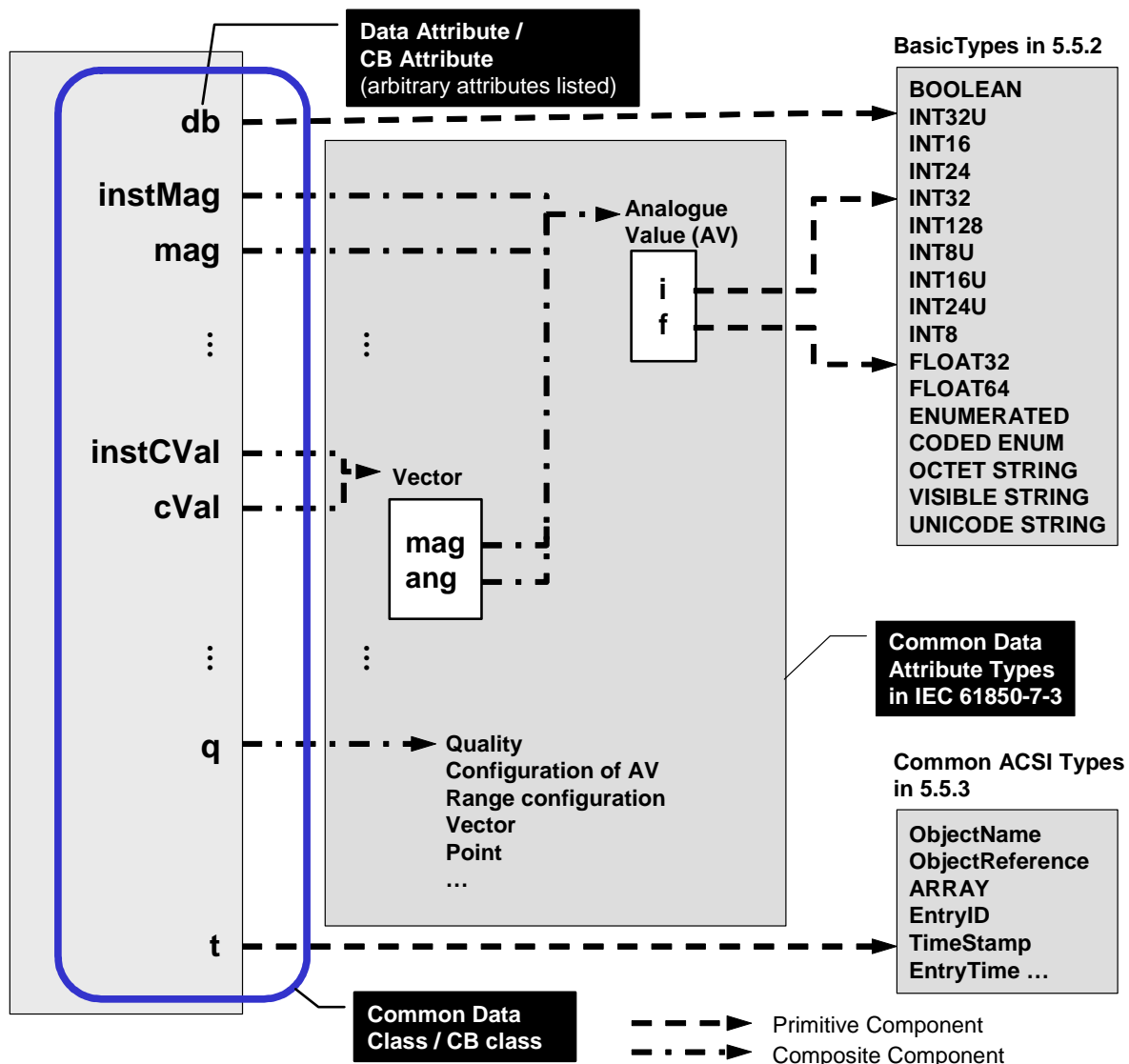
16 **Figure 4 – Data and control blocks (conceptionell model)**

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).



1

2

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

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

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

12 Useful lists of data attributes (components) have been defined for re-use in various application
 13 domains. These lists are called common data classes (control block classes). Common data
 14 classes are defined in IEC 61850-7-3 for utility automation applications and in IEC 61400-25-2
 15 for wind power plant applications. Further application domains may require additional common
 16 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.

4
5

1 **5.5.3 Common ACSI types**

2 **5.5.3.1 General**

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

6 **5.5.3.2 ObjectName**

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

9 **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.			

Add normative text; The constraints in clause 19 on the use of the type ObjectReference shall be applied.

10 ←

11 **5.5.3.3 ObjectReference**

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

16 **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

17

18 The **ObjectReference** syntax shall be:

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

19 The "/" shall separate the instance name of a logical device (**LDName**) from the name of an in-
20 stance of a logical node (**LNName**). The "." shall separate the further names in the hierarchy.
21 The "[.]" shall indicate an option. The inner square bracket "[. ...]" shall indicate further names
22 of recursively nested definitions.

23 NOTE 1 In any case where the context of the text provides sufficient information that an instance of a class is
24 meant, the term "instance of" is not used.

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

← Add normative text; The constraints in clause 19 on the use of the type ObjectReference shall be applied.

1 **5.5.3.4 PHYCOMADDR type**

2 The type **PHYCOMADDR** shall represent a physical communication address (e.g. media ac-
3 cess 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
12 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
17 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.

remove "[" and "]"

add
CDC that does not contain
an ARRAY or , e.g. CMV
in HMV

The definition of Array
needs to say (for e.g.,
HMV, ARRAY OF CMV)
that the definition is
constrained by FC ... we
do not want to have all
Data Attributes, but only
MX ...!!

1 5.5.3.8 Packed list type

2 The **PACKED LIST** type shall be as defined in Table 6.

3 **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

4

5 5.5.3.9 TimeStamp type

6 5.5.3.9.1 General

7 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.

10 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 SCSMs.

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

13 5.5.3.9.2 TimeStamp syntax

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

16 **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 = b ₀ , b ₁ , b ₂ , b ₃ , ...	M
TimeQuality	TimeQuality		M

17

18 5.5.3.9.3 TimeStamp attributes

19 5.5.3.9.3.1 SecondSinceEpoch

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

22 NOTE SecondSinceEpoch corresponds with the Unix epoch.

23 5.5.3.9.3.2 FractionOfSecond

24 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).

27 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.

28

1 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
 2 by an IED is outside the scope of this standard.
 3

4 5.5.3.9.3.3 TimeQuality

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

7 **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

8

1 **LeapSecondsKnown:** The value TRUE of the attribute **LeapSecondsKnown** shall indicate
 2 that the value for SecondSinceEpoch takes into account all leap seconds occurred. If it is
 3 FALSE then the value does not take into account the leap seconds that occurred before the ini-
 4 tialization of the time source of the device.

5 **ClockFailure:** The attribute **clockFailure** shall indicate that the time source of the sending de-
 6 vice is unreliable. The value of the TimeStamp shall be ignored.

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

9 **TimeAccuracy:** The attribute **TimeAccuracy** shall represent the time accuracy class of the
 10 time source of the sending device relative to the external UTC time. The **timeAccuracy**
 11 classes shall represent the number of significant bits in the **FractionOfSecond**.

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

13 NOTE 1 The **TimeAccuracy** meets the requirements specified in IEC 61850-5 for the selected values of n.

14 **Table 9 – TimeAccuracy**

n	Resulting TimeAccuracy (2 ^{**} -n)	Corresponding time performance class de- fined 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)

15

16 5.5.3.10 EntryTime type

17 The type **EntryTime** shall represent the time and date as applied internally for the communica-
 18 tion, reporting, logging, and subsystem as specified by a SCSM.

19 The time base for **EntryTime** shall be GMT. The epoch for EntryTime shall be 01. January
 20 1984 (MJD 40 587).

21 NOTE 1 The TimeStamp type is used for common **DATA** classes in IEC 61850-7-3 and definition of compatible
 22 **DATA** classes in IEC 61850-7-4. The **EntryTime** type is used for all IEC 61850-7-2 class definitions. The **Entry-**
 23 **Time** type may or may not be the same as **TimeStamp** in a SCSM.

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

25 5.5.3.11 TriggerConditions type

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

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

1

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

2

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

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

6 6 SERVER class model

7 6.1 SERVER class definition

8 6.1.1 SERVER class syntax

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

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

13 **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		

14

15 NOTE 2 The server's relationship to the underlying communication system and the concrete implementation de-
16 pend on the SCSM (specific communication service mapping, see IEC 61850-8-x and IEC 61850-9-x) used. Network
17 management (as part of an SCSM), device management, and system management are outside the scope of IEC
18 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 TAppAssociation [0..n] – two-party application association**

13 The attribute **TAppAssociation** 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 MAppAssociation [0..n] – multicast application association**

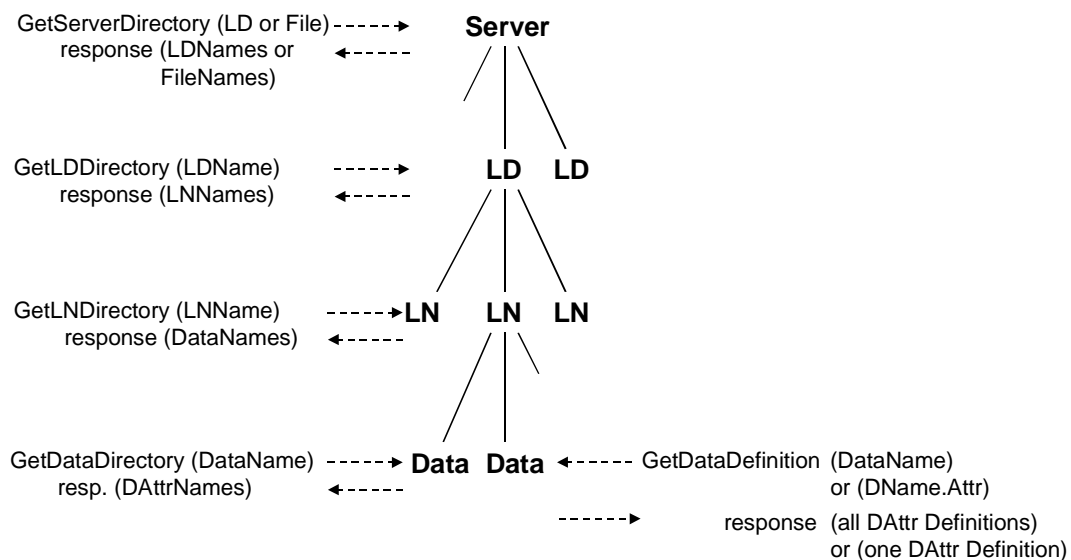
17 The attribute **MAppAssociation** shall identify a subscriber with which a **SERVER** (publisher)
 18 maintains a multicast application association.

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.



1 **Figure 6 – Overview about GetDirectory and GetDefinition services**

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

5 **6.2.2 GetServerDirectory**

6 **6.2.2.1 GetServerDirectory parameter table**

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

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

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

12

13 **6.2.2.2 Request**

14 **6.2.2.2.1 ObjectClass**

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

17 – **LOGICAL-DEVICE**

18 – **FILE**

19 **6.2.2.3 Response+**

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

22 **6.2.2.3.1 Reference [0..n]**

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

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

26 **6.2.2.4 Response–**

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

1 7 Application association model

2 7.1 Introduction

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

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

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

9 NOTE Security requirements are implemented by the SCSMs.

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

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

17 7.2 Concept of application associations

18 The application association model defines

- 19 – the services provided for managing associations between client and server (two-party ap-
20 plication association); and
- 21 – the services provided for managing associations for multicast messaging (for example,
22 GOOSE and transmission of sampled values).

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

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

28 NOTE The details of an application association model are defined in the SCSMs. The following descriptions pro-
29 vide a conceptual model of the application associations between devices.

30 7.3 Access control

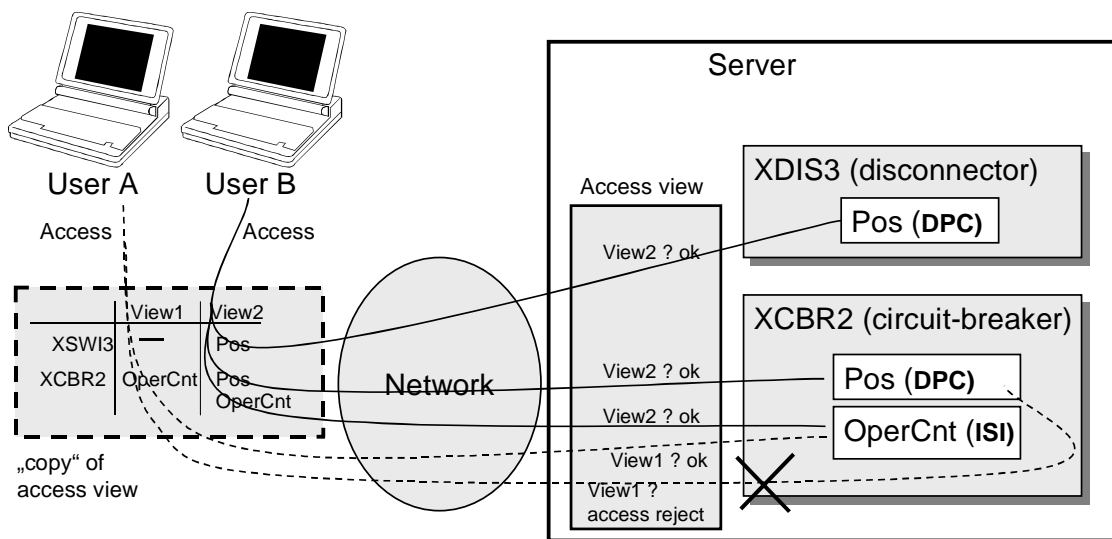
31 The access control model provides the capability to restrict the access of a specific client to
32 class instances, class instance attributes, and ACSI services acting upon class instances of a
33 specific server. The ACSI server contains a set of, for example, **LOGICAL-DEVICES**,
34 **LOGICAL-NODEs**, **DATA**, or report controls. The set of instances visible (and therefore ac-
35 cessible) to a client is restricted on the basis of the control definition of the server. This restricted set
36 of instances visible to a client may not only restrict the visibility of instances but also the access
37 to the service. The concept of a virtual access view is illustrated in Figure 1.

38 NOTE 1 The virtual access view is the authentication's view of the server.

Revise and take the
role based access
concept of part 6 into
account.

Who does the specification?

- Document from WWs
- WW used input KHS
- What needs to be defined in 7-2?
- Services need t have TLS or something like that
- It can be restricted to a LNTtype in SCL
- list of standardized roles required (...)
- Can we define roles to be FC sensitiv? Yes!
- A DATaSet shall not have members from different roles!!
- need to coordinate with WG 15
- keep the general model description and formal model in one document as we did for SHS.



IEC 401/03

1
2 **Figure 7 – Access views of a server**

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

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

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

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

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

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

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

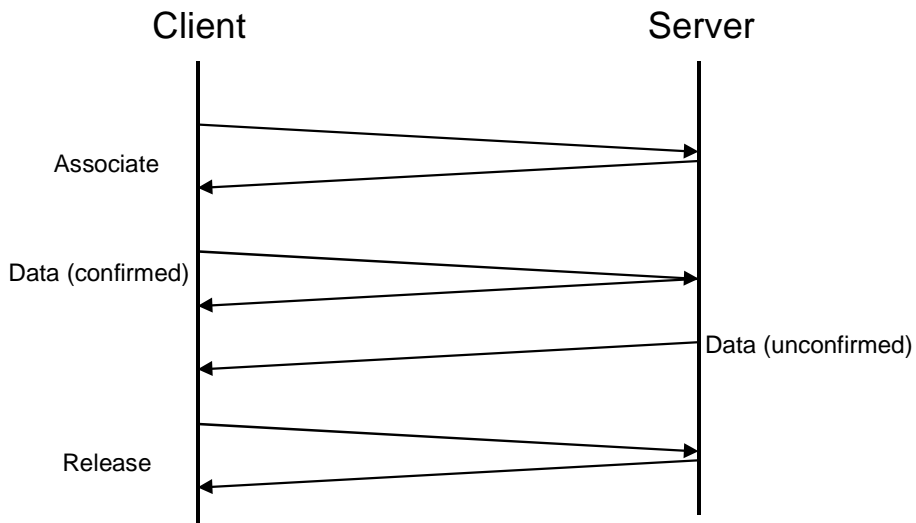
21 **7.4 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model**

22 **7.4.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition**

23 **7.4.1.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class syntax**

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

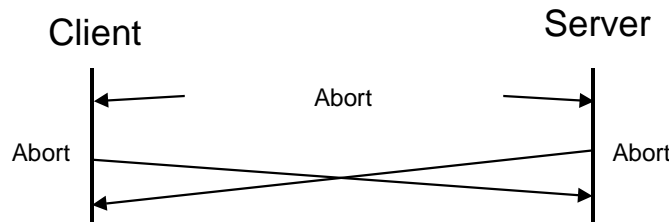
1 The services for associate, data exchange, and association release of the two-party application
 2 association class is depicted in Figure 8.



IEC 402/03

3
 4 **Figure 8 – Normal operation**

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



IEC 403/03

6
 7 **Figure 9 – Aborting association**

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

9 **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")		

10

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 **ServiceError**
16 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.

1 7.5 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class

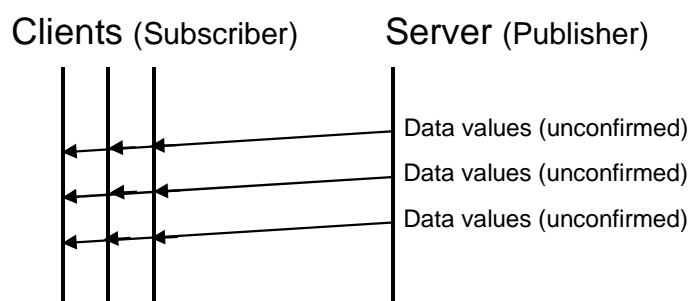
2 7.5.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition

3 7.5.1.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class syntax

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

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

13 The multicast application association class is depicted in Figure 10.



14
15 **Figure 10 – Principle of multicast application association**

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

17 **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")		

18

19 7.5.1.2 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class attributes

20 7.5.1.2.1 AuthenticationParameter

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

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

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

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

1 8 LOGICAL-DEVICE class model

2 8.1 LOGICAL-DEVICE class definition

3 8.1.1 LOGICAL-DEVICE class syntax

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

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

8 **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		

9

10 8.1.2 LOGICAL-DEVICE class attributes

11 8.1.2.1 LDName – logical device name

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

← subnetwork

14 8.1.2.2 LDRef – logical device ObjectReference

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

LDName

16

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

19 8.1.2.3 LogicalNode [2..n]

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

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

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

1 8.2 LOGICAL-DEVICE class services

2 8.2.1 GetLogicalDeviceDirectory

3 8.2.1.1 GetLogicalDeviceDirectory parameter table

4 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**.

7 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

9

10 8.2.1.2 Request

11 8.2.1.2.1 LDReference – logical device ObjectReference

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

14 8.2.1.3 Response+

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

17 8.2.1.3.1 LNReference [3..n] – logical node ObjectReference

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

20 8.2.1.4 Response–

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

23

1 9 LOGICAL-NODE class model

2 9.1 LOGICAL-NODE class definition

3 9.1.1 LOGICAL-NODE class syntax

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

6 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 LNN		
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 following attributes shall only be available if their support is explicitly stated in the definition of a compatible LN class, e.g., in IEC 61850-7-4.

Zug, 2007-07-10

7

8 The definition of **LOGICAL-NODEs** for the utility automation domain is refined by the definition
9 of specific **DATA** in IEC 61850-7-4. The definitions in IEC 61850-7-4 (and IEC 61850-7-3 for
10 the common **DATA** classes) shall be taken into account to get the comprehensive definition of
11 utility automation-domain-specific **LOGICAL-NODEs**.

12 NOTE 2 IEC 61850-7-4 defines further attributes for **LOGICAL-NODEs**; for example., the mode (behaviour: ON,
13 BLOCKED, TEST, etc.) of the utility automation-specific **LOGICAL-NODE** is defined in IEC 61850-7-4. The state
14 model of a **LOGICAL-NODE** is modelled as a specific **DATA** (named **Mod**).

15 9.1.2 LOGICAL-NODE class attributes

16 9.1.2.1 LNName – Logical node name

17 The attribute **LNName** shall unambiguously identify **LOGICAL-NODE** within the scope of
18 **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

a) **GetLogicalNodeDirectory(with ACSIClass == DATA)** shall return one Data Level (this is what the LN tables in IEC 61850-7-4 or IEC 61400-25-2 show).

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.

1 **10 DATA class model**

2 **10.1 General**

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

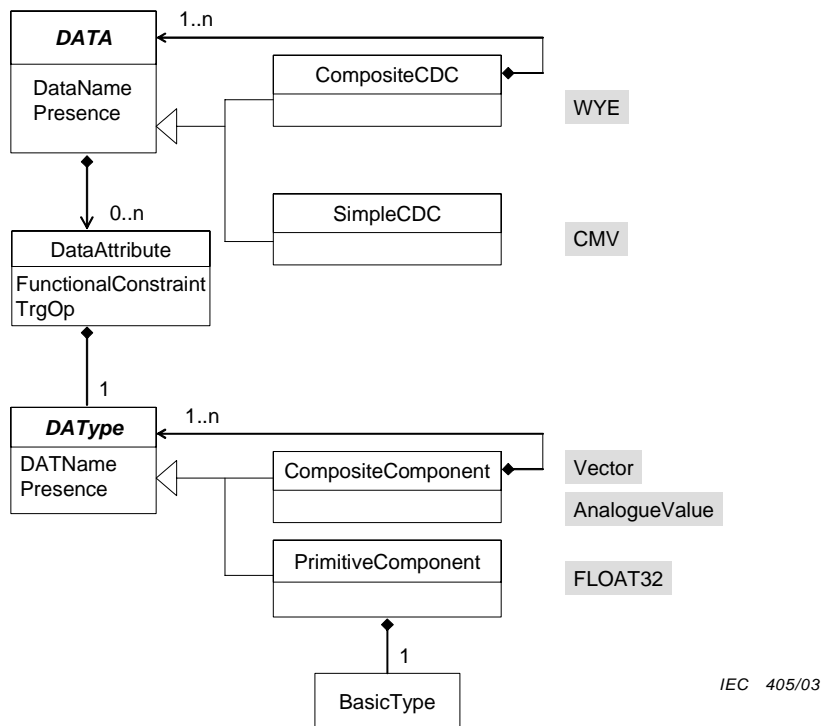
12 NOTE 1 The consequences of setting values to instances of **DATA** is outside this part of IEC 61850. IEC 61850-7-3
 13 and IEC 61850-7-4 define many utility-domain-specific **DATA** classes. These definitions provide information on the
 14 actions to be taken by the receiving application, for example,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

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

20 **10.2 DATA class definition**

21 **10.2.1 DATA class syntax**

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



24

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

26 NOTE 1 The example in Figure 11 uses definitions (for example, **WYE**, **CMV**, **Vector**, and **AnalogueValue** com-
 27 mon **DATA** classes) found in IEC 61850-7-3. A comprehensive introduction to the modelling of **DATA** can be found
 28 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**.

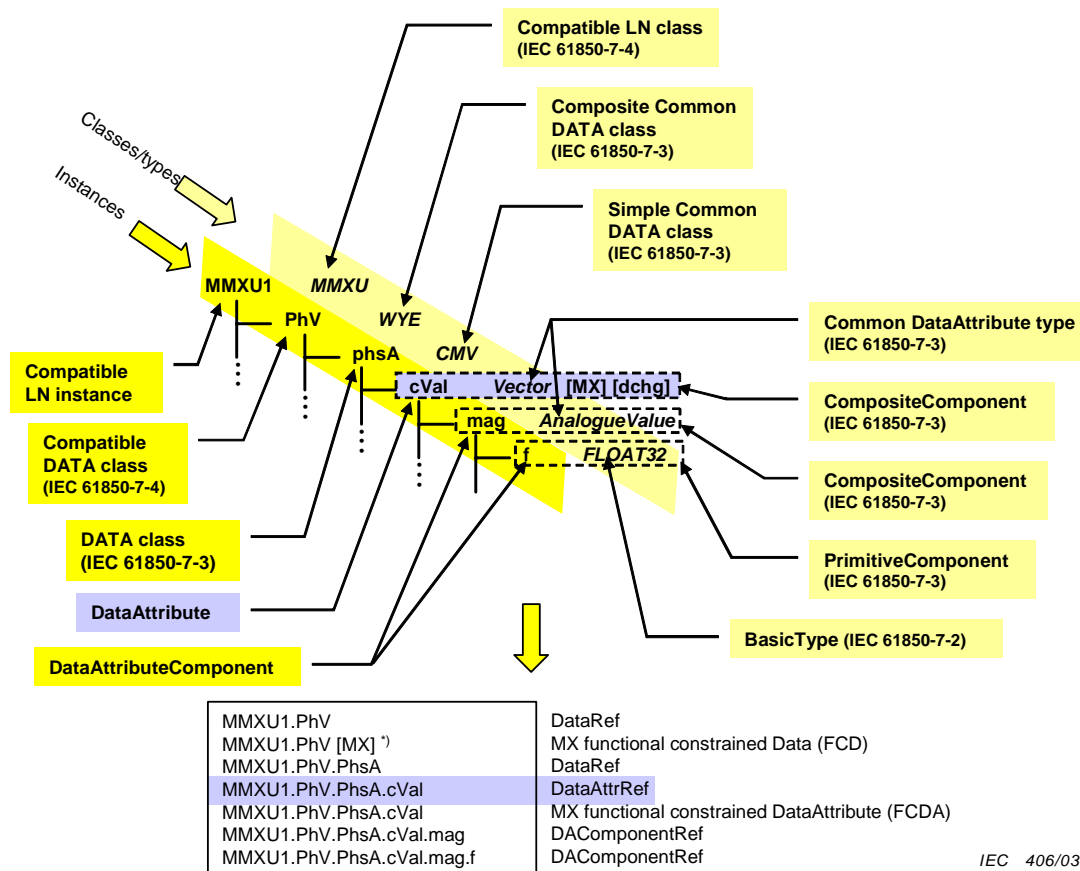
3 NOTE 2 The **DATA** class is an abstract class that is an auxiliary means to construct the primitive and composite
 4 common data classes.

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 **DataAttribute**
 19 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.

21

22

Figure 12 – Example of DATA

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

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

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

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

8 The table notation does not easily show the inheritance. Therefore the class diagram in Figure
9 11 shall be normative. The tables and the class diagrams shall be used together.

10 **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		

11
12 An instance of a **DATA** class may contain zero or more instances of a **CompositeCDC**, **SimpleCDC**
13 or a **DataAttribute**. However, they cannot all be absent, so at least one of these ele-
14 ments shall be present.

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

18 NOTE 6 **DATA** or part of a **DATA** may be referenced in a **DATA-SET**. The persistent existence of **DATA** is ex-
19 pected as long as they are referenced as members of a **DATA-SET**. A system has to take special measures to en-
20 sure their existence.

21 10.2.2 DATA class attributes

22 10.2.2.1 DataName

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

25 10.2.2.2 DataRef – data ObjectReference

26 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 conceptual 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**.

The **DATName** (if **DataAttribute** is not nested) or the **DATName** of the first level (if Data Attribute is nested) shall be called the **DataAttributeName**.

For the second and any deeper nesting levels the **DATName** shall be called **DAComponentName**.

The **ObjectReference** from the top (Logical Device) down to the **DataAttributeName** shall be called **DataAttributeReference**.

1 EXAMPLE As shown in Figure 12, the **cVal** (derived from a common data attribute type – **Vector**) is
 2 the DataAttribute. The **mag** (also derived from a common data attribute type – **AnalogueValue**) is a DataAt-
 3 tributeComponent.

4 **DATRef – data attribute type ObjectReference**

5 The attribute **DATRef** shall be the unique path-name of a **DAType**

6 The ObjectReference **DATRef** shall be:

LDName/LNName.
DataName[.DataName[. ...]].DataAttributeName[.DACom

7

8 The ObjectReference **DataAttributeReference** shall be:

LDName/LNName.
DataName[.DataName[. ...]].DataAttributeName[NumArrayElement]

9

10 NOTE 3 Nesting depends on the concrete definition of a **DATA** class and **DAType** class.

11 NOTE 4 In each path within a **DATA** there is one and only one **DataAttribute** (level). Only **DataAttributes** have
 12 the functional constraint (FC) and trigger option (TrgOp).

13 The NumArrayElement (value between 0 and m according to the instance of array) shall be the
 14 optional array element number in case a single array element is to be

15 **Presence**

16 The attribute **Presence** of type **BOOLEAN** shall define if a **Data** is mandatory (Presence = TRUE) or
 17 optional (Presence = FALSE).

18 NOTE The Presence attribute is a conceptual attribute only. The common data classes as defined, e.g., in IEC
 19 61850-7-3 use a column "M/O/C" (mandatory / optional / conditional) to represent if a data attribute or common data
 20 attribute shall be present in an implementation.

21 **CompositeComponent [0..n] – composite component**

22 The attribute **CompositeComponent** shall be a specialization of **DAType**.

23 **PrimitiveComponent [0..n] – primitive component**

24 The attribute **PrimitiveComponent** shall be a specialization of **DAType**.

replace "[" by "[" and "]" by "]". The "()" means Arrayelement, the "[]" means optional. BUT: this cannot be mapped to any MMS ObjectName, the ObjectNames have a restricted character set according to clause 19.
 Discussion:
 ArrayElement can also be at lowest DATA level, see HMV, ... and this cannot be mapped to DatRef in Reports and Logs!
 Cannot define it in a DataSet, because FCD/FCDA does not allow to describe an array element XXX (3).
 Need to add in FCDA the "(Arraynumber)"

Currently:
 (Arrayelementnumber)
 can be on several levels (not only at Dataattribute"; change this in 5.5.3.5

1 **10.2.2.4.2 FunctionalConstraint [1..2] – functional constraint**

2 From an application point of view, the **DataAttributes** are classified according to their specific
3 use; for example, some attributes are used for **controlling** purposes, other attributes are used
4 for **reporting** and **logging**, **configuration**, others indicate **measurements** or **setting groups**,
5 or some identify the **description** of a specific **DataAttribute**.

6 The **FunctionalConstraint (FC)** shall be a property of the **DataAttribute** characterizing the
7 specific use of the **DataAttribute**. The **FunctionalConstraint** is used in the definition of
8 **DATA** (contained in **LOGICAL-NODES**) and of the various control blocks (for example, **BRCB**).
9 Most attributes of control blocks have a **FunctionalConstraint** property.

10 NOTE The **FunctionalConstraint** could be understood as a filter of the **DataAttributes**. The common data classes
11 in IEC 61850-7-3 use the **FunctionalConstraint** values defined in this subclause.

12 The **FunctionalConstraint** is used in various definitions in this part of IEC 61850. The **Func-**
13 **tionalConstraint** shall indicate the services that are allowed to be operated on a specific
14 **DataAttribute**. **The FunctionalConstraints shall be as specified in Table 18.**

15 **Table 18 – Functional constraint values**

Make two tables: one for FCD/FCDA and one for the new FCCB (for control blocks)

FunctionalConstraint values					
FC	Semantic	Services allowed	Initial values/storage/ explanation	D ^a	CB ^b
ST	Status information	DataAttribute shall represent a status information whose value may be read, substituted, reported, and logged but shall not be written	Initial value of the DataAttribute shall be taken from the process	X	
MX	Measurands (analogue values)	DataAttribute shall represent a measurand information whose value may be read, substituted, reported, and logged but shall not be written	Initial value of the DataAttribute shall be taken from the process	X	
CO	Control	DataAttribute shall represent a control information whose value may be operated (control model) and read	N.a.	X	
SP	Setpoint	DataAttribute shall represent a setting parameter information whose value whose value is read and may be written. Changes of values shall become effective immediately	Initial value of the DataAttribute shall be as configured; value shall be non-volatile	X	X
SV	Substitution	DataAttribute shall represent a substitution information whose value may be written to substitute the value attribute and read	If the value of the DataAttribute is volatile then the initial value shall be FALSE, else the value should be as set or configured	X	
CF	Configuration	DataAttribute shall represent a configuration information whose value may be written and read. Values written may become effective immediately or deferred by reasons outside the scope of this standard	Initial value of the DataAttribute shall be as configured; value shall be non-volatile	X	
DC	Description	DataAttribute shall represent a description information whose value may be written and read	Initial value of the DataAttribute shall be as configured; value shall be non-volatile	X	
SG	Setting group	Logical devices that implement the SGCB class maintain multiple grouped values of all instances of DataAttributes with functional constraint SG. Each group contains one value for each DataAttribute with functional constraint SG which shall be the current active value (for details see 13). Values the of DataAttributes with FC=SG shall not be writeable	Initial value of the DataAttribute shall be as configured; value shall be non-volatile	X	
SE	Setting group editable	DataAttribute which can be edited by SGCB services	Value of the DataAttribute shall be as available after Select-EditSG service has been proc-	X	

FunctionalConstraint values					
FC	Semantic	Services allowed	Initial values/storage/ explanation	D ^a	CB ^b
			essed		
EX	Extended definition	DataAttribute shall represent an extension information providing a reference to a name space. Extensions are used in conjunction with extended definitions of LN s, DATA , and DataAttributes in 61850-7-3 and IEC 61850-7-4. Values the of DataAttributes with FC=EX shall not be writeable	Value of the DataAttribute shall be as configured; value shall be non-volatile	X	
BR	Buffered report ^c	Attribute shall represent a report control information of a BRCB whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
RP	Unbuffered report ^c	Attribute shall represent a report control information of a URCB whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
LG	Logging ^c	Attribute shall represent a log control information of a LCB whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
GO	Goose control ^c	Attribute shall represent a goose control information of a GoCB whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
GS	Gsse control ^c	Attribute shall represent a goose control information of a GsCB whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
MS	Multicast sampled value control ^c	Attribute shall represent a sampled value control information of a MSVCB whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
US	Unicast sampled value control ^c	Attribute shall represent a sampled value control information of an instance of a UNICAST-SVC whose value may be written and read	Initial value of the Attribute shall be as configured; value shall be non-volatile		X
XX	Representing all DataAttributes as a service parameter	Shall represent all DataAttributes of a DATA (of any FC) to be accessed, for example, to be written and read. The FC value "xx" shall only be used in the functional constrained data (FCD); "XX" shall not be used as FC value in a DataAttribute	"XX" shall be used as a wildcard in services only		
NOTE The possibility to write an Attribute or a DataAttribute may be further constrained by a view or an implementation.					
^a Column D indicates the use of the FC in the definition of DATA (i.e. common DATA classes in IEC 61850-7-3).					
^b Column CB indicates the use of the FC in the definition of control blocks in this part of IEC 61850.					
^c Reserved for control classes in this part of IEC 61850.					

1
 2 EXAMPLE The common data attribute class **single-point status (SPS)** according to
 3 IEC 61850-7-3 has the following **Data** **CR Control response** (value), **q** (quality), and **t** (time stamp) with the
 4 functional constraint **ST** (status information).
 5 **10.2.2.4.3 TrgOp [0..1] – trigger option**
 6 The attribute **TrgOp** of type **TriggerConditions** (see Table 10; only the first, third and fourth
 7 columns are relevant) shall define the trigger conditions (associated with a **DataAttribute** of a
 8 **DATA**) that may cause a report to be sent or a log entry to be stored into a log (report model;
 9 see Clause 14). The services associated with the **TrgOpt** shall be as specified in Table 19.

CN Control negative response
 CR Control response
 SCL Control (8-1) ??

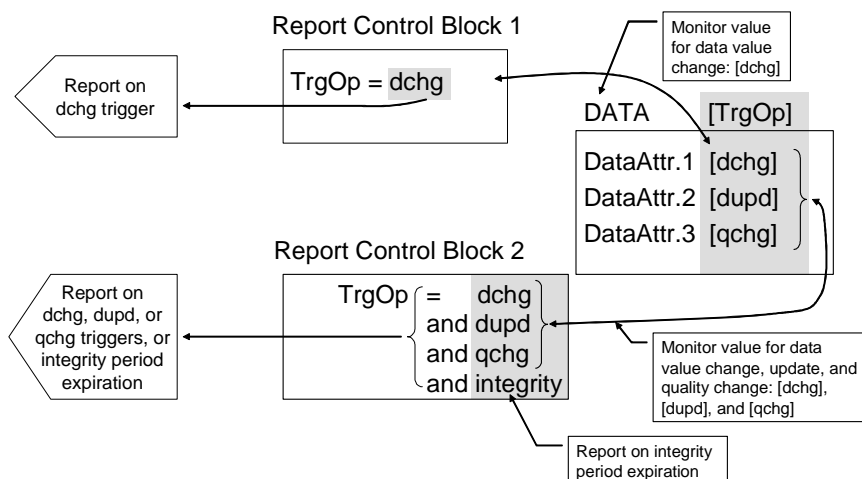
1

Table 19 – TrgOp

TrgOp	Semantic	Services allowed
dchg	data-change	A report or a log entry shall be generated due to a change of the value of the associated DataAttribute
qchg	quality-change	A report or a log entry shall be generated due to a change of the value of the associated quality DataAttribute q
dupd	data value update	A report or a log entry shall be generated due to freezing the value of a freezable DataAttribute or updating the value of any other DataAttribute . An updated value may have the same value as the old value

NOTE The trigger conditions integrity and general-interrogation of the TriggerConditions type (see Table 10) are used independent of instances of **DATA**; they can be set from remote by services and thus trigger sending reports or placing log entries into logs. If a **DataAttribute** is a composite component, the change or update of a **DataAttribute** shall be understood as the change or update of one or more of the primitive components of the **DataAttribute**.

2 As depicted in Figure 13 the value of a **DataAttribute** that provides a specific **TrgOp** (trigger
 3 option) shall be monitored for reporting and logging if the report control block has enabled the
 4 specific trigger option (**TrgOp**). In the upper example of Figure 13 the **TrgOp** is **dchg**; the
 5 **TrgOp** of the **DataAttributes** is **dchg** for the first, **dupd** for the second, and **qchg** for the last
 6 **DataAttribute**. Reports are sent on data changes only, because only **dchg** is enabled in the
 7 report control block. In the second example, all changes will be reported. In addition, a report
 8 will be sent on the expiration of the integrity period.



9

Figure 13 – Relation of TrgOp and Reporting

IEC 407/03

10

11 **DATA** whose **DataAttributes** shall be monitored for change detection shall be referenced by
 12 a **DATA-SET**.

13 **EXAMPLE** Common data attributes in IEC 61850-7-3, for example, **stVal** (status value) provides a trigger option
 14 **dchg**, the common data attribute **q** (quality) provides the trigger option **qchg**.

15 **NOTE** The data attributes of **DATA-SET** which will be reported or logged after a change has been detected de-
 16 pend on the definition of the data set used for reporting. For details see Clause 11.

17 **10.2.2.4.4 Functional constrained data (FCD)**

Introduce the FCCB (Functional Constraint for Control Blocks) from Part 6 Ed2

18 The reference of an ordered collection of **DataAttributes** of a **DATA** (or attributes of a control
 19 block) having the same **functional constraint (FC)** value shall be called **functional con-**
 20 **strained data (FCD)**. The order of the collection of the **FCD** shall be the order of the appear-
 21 ance of the **DataAttributes** in the **DATA** (or attributes of a control block).

1 NOTE All measured values of a **DATA (FC = MX)** are referenced by the **measurement FCD**. The functional con-
 2 strained data is used, for example, to describe and to remotely create **DATA-SETS**. The syntax notation for **FCD** is
 3 defined in a SCSM.

4 EXAMPLE Figure 12 shows a **[MX] FCD** in the second line in the box of the bottom.

5 **10.2.2.4.5 Functional constrained data attribute (FCDA)**

Should allow also subsets of DataAttributes! as shown in Fig 15.

6 A reference of a single **DataAttribute** of a **DATA** (or attributes of a control block) having a
 7 specific **functional constraint (FC)** value shall be called **functional constrained data at-**
 8 **tribute (FCDA)**.

A reference of a single data attribute of a data or data attribute component of a data attribute or a attribute of a control block having a specific functional constraint value shall be called functional constrained data attribute.

9 NOTE A single meas
 10 attribute is used, for example, to describe and to remotely create **DATA-SETS**. The syntax notation for **FCDA** is de-
 11 fined in a SCSM.

12 EXAMPLE Figure 12 shows a **[MX] FCDA** in the fifth line in the box of the bottom.

Need to add FCCB - functional constrained control block attributes. This has been added in part 6 (SCL)

13 **10.2.2.5 CompositeCDC [0..n]**

14 The attribute **CompositeCDC** shall be a specialization of **DATA**.

15 **10.2.2.6 SimpleCDC [0..n]**

16 **10.2.2.6.1 SimpleCDC syntax – General**

17 The attribute **SimpleCDC** shall be a specialization of **DATA**.

18 **10.2.2.6.2 COMMON-DATA class syntax**

19 The **COMMON-DATA** class shall be as defined in Table 20.

20 **Table 20 – COMMON-DATA class definition**

COMMON-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 [1..n] DataAttributeType FunctionalConstraint TrgOp [0..n]	DAType FC TriggerConditions	For example, Vector class of IEC 61850-7-3 for example, MX for example, dchg
Services GetDataValue SetDataValue GetDataDirectory GetDataDefinition		
NOTE 1 The CommonDATA is a subclass of the DATA class . NOTE 2 DATA or DataAttribute may be referenced in a DATA-SET . The persistent existence of DATA and DataAttribute 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. NOTE 3 IEC 61850-7-2 defines the basic class model. IEC 61850-7-3 defines specialized DATA classes – the common DATA classes, for example, SPS modelling a single-point status DATA class. IEC 61850-7-4 defines specialized common DATA classes – the compatible DATA classes, for example, Pos modelling a position (specializing a SPS common DATA class).		

Reading array elements is not possible?
 Should allow "()" in the services.
 Mapping needs to be modified.

1 **DataName**

2 The attribute **DataName** shall identify a **DATA** within the scope of a **LOGICAL-NODE** or a
3 nested **DATA**.

4 **DataRef – data ObjectReference**

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

6 The ObjectReference **DataRef** shall be:

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

7

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

9 **Presence**

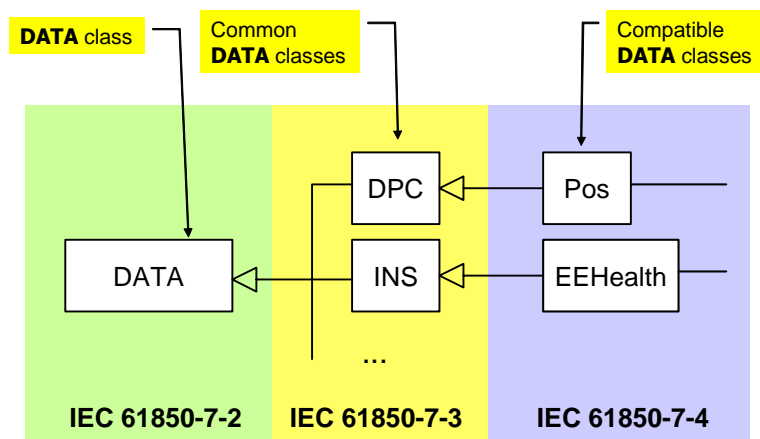
10 The attribute **Presence** of type **BOOLEAN** shall define if a **DATA** is mandatory (**Presence =**
11 **TRUE**) or optional (**Presence = FALSE**).

12 **DataAttribute**

13 The attribute **DataAttribute** shall be as defined in 10.2.2.4.

14 **10.3 Relation of DATA, common DATA, and compatible DATA classes**

15 The **DATA** defines a class that is specialized in IEC 61850-7-3 to define the common **DATA**.
16 IEC 61850-7-4 specializes the common **DATA** class (to define the compatible **DATA** class).
17 The relation between these parts is depicted in Figure 14.



IEC 408/03

18

19

Figure 14 – Relation of DATA classes

20 NOTE The common **DATA** class in IEC 61850-7-3 “adds” common structures (the **DataAttributes**) to the **DATA**
21 class; the compatible **DATA** class in IEC 61850-7-4 “adds” specific semantic to a specialized common **DATA** class.

22 EXAMPLE The compatible **DATA** class with the name “**Pos**” represents a switch position. “**Pos**” is a specialization
23 of the common **DATA** class “**DPC**” (double-point control). The **DATA** “**Pos**” may be used in one or several
24 **LOGICAL-NODEs**.

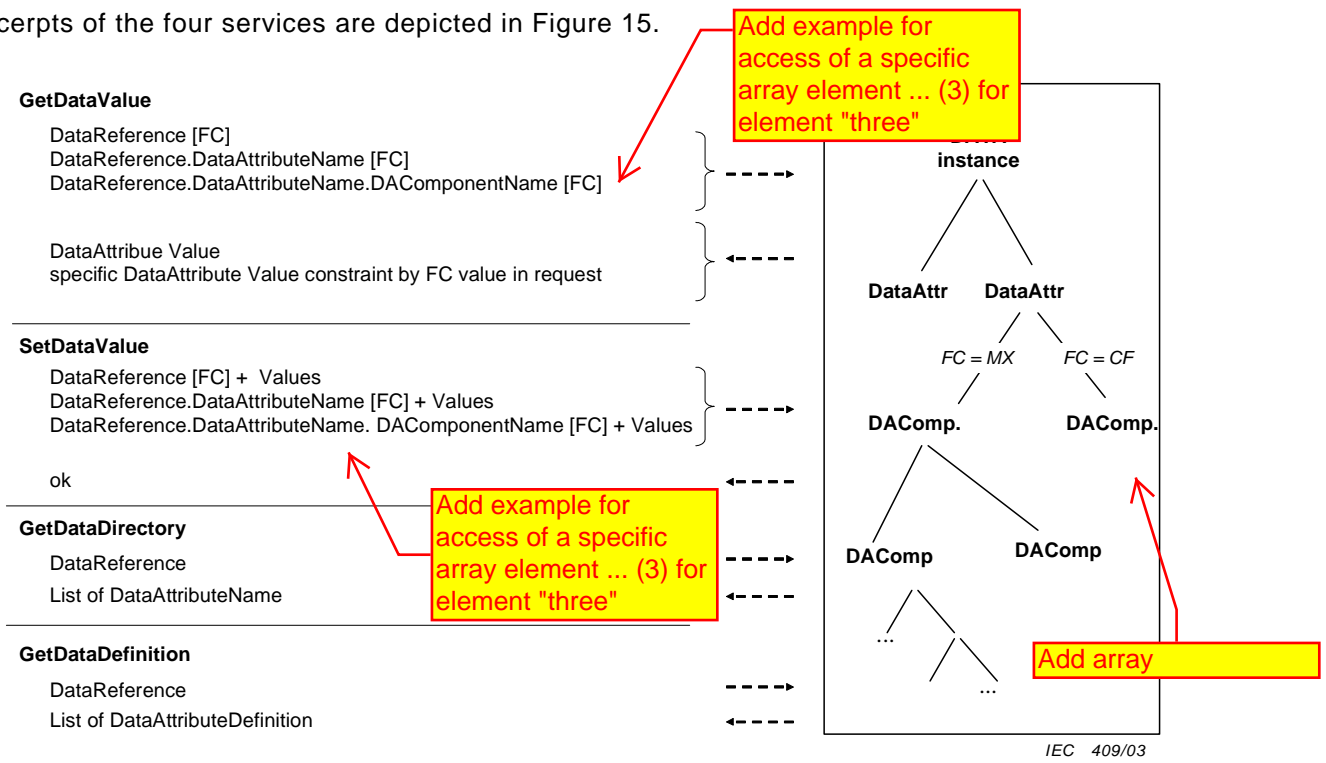
1 **10.4 DATA class services**

2 **10.4.1 General definitions and overview**

3 For **DATA** the following services are defined.

Service	Description
GetDataValue	Retrieve value of DATA contained in the LOGICAL-NODE
SetDataValue	Write value of DATA contained in the LOGICAL-NODE
GetDataDirectory	Retrieve ObjectReferences of all DataAttributes contained in the DATA
GetDataDefinition	Retrieve definitions of all DataAttributes contained in the DATA

4 Excerpts of the four services are depicted in Figure 15.



5
6

7 **Figure 15 – Excerpt of data class services**

8 The **GetDataValue** and **SetDataValue** services allow accessing a complete **DATA** or any part
 9 of it.

10 **10.4.2 GetDataValue**

11 **10.4.2.1 GetDataValue parameter table**

12 A client shall use the **GetDataValue** service to retrieve value of **DataAttributes** of the refer-
 13 enced **DATA** made visible and thus accessible to the requesting client by the referenced
 14 **LOGICAL-NODE**.

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

Parameter name
Request
Reference
Response+
DataAttributeValue [1..n]
Response–
ServiceError

1

2 **10.4.2.2 Request**3 **10.4.2.2.1 Reference**

4 The parameter **Reference** shall define the **functional constrained data (FCD)** or **functional constrained data attributes (FCDA)** of the **DATA** whose **DataAttribute** values are
5 to be retrieved. The **Reference** shall be **FCD** or **FCDA**.
6

7 NOTE An SCSM may provide access to a range of ARRAY elements or a single ARRAY element.

8 **10.4.2.3 Response+**

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

11 **10.4.2.3.1 DataAttributeValue [1..n]**

12 The parameter **DataAttributeValue** shall contain

- 13 – the values of all **DataAttributes** of a **DATA** referenced by **FCD**; or
- 14 – the value of a **DataAttribute** referenced by **FCDA**.

15 NOTE The syntax of the **DataAttributeValue** is defined in an SCSM.

16 **10.4.2.4 Response–**

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

19 **10.4.3 SetDataValue**20 **10.4.3.1 SetDataValue parameter table**

21 A client shall use the **SetDataValue** service to set value of **DataAttributes** of the referenced
22 **DATA** made visible and thus accessible to the requesting client by the referenced **LOGICAL-**
23 **NODE**.

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

Parameter name
Request
Reference
DataAttributeValue [1..n]
Response+
Response–
ServiceError

1

2 **10.4.3.2 Request**3 **10.4.3.2.1 Reference**

4 The parameter **Reference** shall define the **functional constrained data (FCD)** or **func-**
 5 **tional constrained data attributes (FCDA)** of the **DATA** whose **DataAttribute** values are
 6 to be set. The **Reference** shall be **FCD** or **FCDA**.

7 NOTE An SCSM may provide access to a range of ARRAY elements or a single ARRAY element.

8 **10.4.3.2.2 DataAttributeValue [1..n]**

9 The parameter **DataAttributeValue** shall contain

- 10 – the values of all **DataAttributes** of a **DATA** referenced by **FCD**; or
- 11 – the value of a **DataAttribute** referenced by **FCDA**.

12 NOTE The syntax of the **DataAttributeValue** is defined in an SCSM.

13 **10.4.3.3 Response+**

14 The parameter **Response+** shall indicate that the service request succeeded. The type for this
 15 parameter is SCSM specific.

16 NOTE 1 For the **SetDataValue** service, a successful result means that the service request was acceptable to the
 17 server and that the server has attempted to move the value of each **DataAttribute** of the **DATA** requested by the
 18 service to the corresponding application.

19 NOTE 2 The action to be taken by an application receiving the value for a **DATA** to be set is outside the scope of
 20 this standard.

21 **10.4.3.4 Response–**

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

24 The attempt to set a DataAttribute or an underlying component that is not available shall be in-
 25 terpreted as a service failure.

26 **10.4.4 GetDataDirectory**27 **10.4.4.1 GetDataDirectory parameter table**

28 A client shall use the **GetDataDirectory** service to retrieve the list of all **DataAttribu-**
 29 **teNames** of the referenced **DATA** made visible and thus accessible to the requesting cli-
 30 ent by the referenced **LOGICAL-NODE**.

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

Parameter name
Request
DataReference
Response+
DataAttributeName [1..n]
Response–
ServiceError

GetDataDirectory will remain and it shall return the list of ObjectNames of the elements that compose the ObjectReference sent in the service request. This means the top level DataNames and DataAttributeNames. (this is what the CDC tables in IEC 61850-7-3 show)

3

4 10.4.4.2 Request

5 DataReference – data reference

6 The parameter **DataReference** shall contain the **ObjectReference** of a **DATA**. The **Ob-**
7 **jectReference** shall be **DataRef**.

8 10.4.4.3 Response+

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

11 DataAttributeName [1..n]

12 The parameter **DataAttributeName** shall contain a **DataAttrName** of the highest level of a
13 **DataAttribute** of the **DATA**.

14 10.4.4.4 Response–

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

17 10.4.5 GetDataDefinition

18 10.4.5.1 GetDataDefinition parameter table

19 A client shall use the **GetDataDefinition** service to retrieve the complete list of all **DataAt-**
20 **tribute** definitions of the referenced **DATA** made visible and thus accessible to the requesting
21 client by the referenced **LOGICAL-NODE**.

22 NOTE 1 Complete means that the whole structure (the tree with all its branches and leaves) of each **DataAttrib-**
23 **ute** shall be retrieved, i.e., all nested DataAttribute.

24 NOTE 2 The visible instances are those that are defined within a given view (see Clause 7 for details on
25 the view concept).

Parameter name
Request
DataReference
Response+
DataAttributeDefinition [0..n]
Response–
ServiceError

GetDataDirectory will remain and it shall return the list of ObjectNames of the elements that compose the ObjectReference sent in the service request. This means the top level DataNames and DataAttributeNames. (this is what the CDC tables in IEC 61850-7-3 show)

26

1 **10.4.5.2 Request**

2 **DataReference – data ObjectReference**

3 The parameter **DataReference** shall contain the **ObjectReference** of the **DATA**. The **Objec-**
4 **tReference** shall be **DataRef**.

5 NOTE An SCSM may bundle several **DataReference** parameters into one message.

6 **10.4.5.3 Response+**

7 **DataAttributeDefinition [0..n]**

8 The parameter **DataAttributeDefinition** shall contain a **DataAttrName** and **DataAttrType**
9 of the first level and of all nested levels below of the referenced **DATA** and the functional con-
10 straints of each **DataAttribute** where applicable.

11 **10.4.5.4 Response–**

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

14

1 11 DATA-SET class model

2 11.1 General

3 A **DATA-SET** is an ordered group of **ObjectReferences** of **DATA** or **DataAttributes** (called
4 the data set members), organized as a single collection for the convenience of the client. The
5 membership and order of the **ObjectReferences** in a **DATA-SET** shall be known to both
6 the client and the server, so that only the name of the **DATA-SET** and the current values of the
7 referenced **DATA** or **DataAttributes** need to be transmitted. This capability thus permits more
8 efficient use of the communications bandwidth.

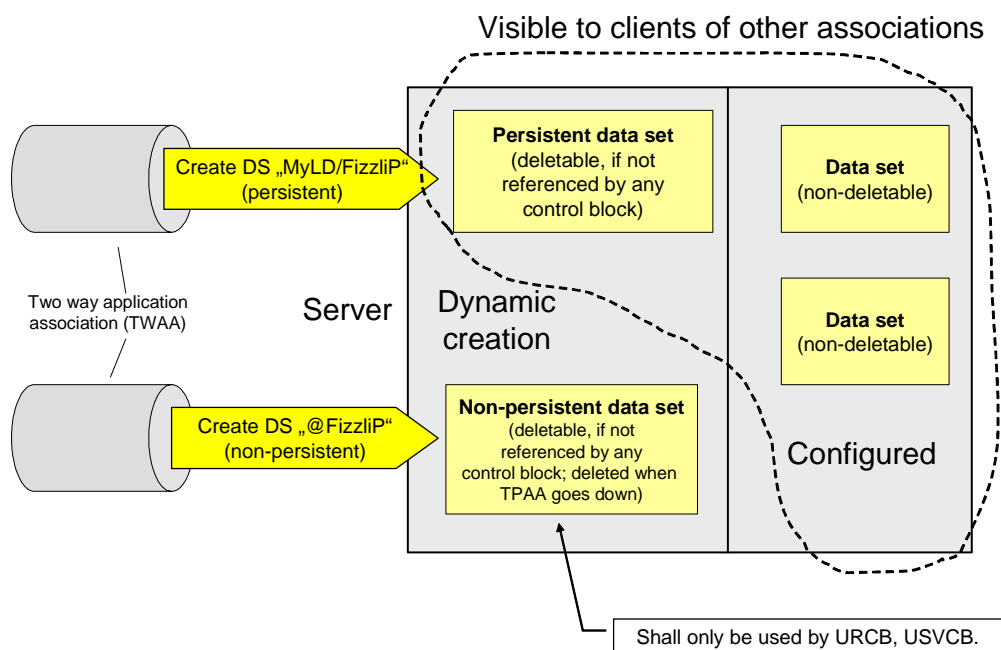
9 NOTE 1 The membership and order of the **DATA** or **DataAttribute** in a **DATA-SET** can be retrieved with the
10 **GetDataSetDirectory** service. The persistent existence of **DATA** and **DataAttribute** is expected as long as they
11 are referenced as members of a **DATA-SET**. A system has to take special measures to ensure their persistent exist-
12 tence.

13 **DATA-SETS** are also important for control models, for example, reporting, logging, **GOOSE**.
14 **DATA-SETS** are used, for example, to define the values of **DATA** or **DataAttributes** to be
15 transmitted in case of a value change of one of its members.

16 **DATA-SETS** may be configured or created through the **CreateDataSet** service.

17 Any **DATA** or **DataAttributes** in a **SERVER** may be referenced by one or more **DATA-SETS**.

18 A **DATA-SET** may be created through the **CreateDataSet** service as a persistent or a non-
19 persistent instance of **DATA-SET** (see Figure 16). A persistent instance of **DATA-SET** shall be
20 visible to clients of any **TWO-PARTY-APPLICATION-ASSOCIATION**. Non-persistent in-
21 stances shall be visible only to the client that created the instance. Pre-defined (configured) in-
22 stances of **DATA-SET** shall be visible to clients of any **TWO-PARTY-APPLICATION-**
23 **ASSOCIATION** and they shall be non-deletable.



24

25

Figure 16 – Dynamic creation of data set instances

IEC 410/03

26 Persistent instances of **DATA-SETS** shall **not be deleted** when the **TWO-PARTY-**
27 **APPLICATION-ASSOCIATION** over which the instance has been created is released or

1 aborted. Non-persistent instances shall be **automatically deleted** when the **TWO-PARTY-**
 2 **APPLICATION-ASSOCIATION** over which the instance has been created is released or
 3 aborted. Persistent **DATA-SETs** created through the **CreateDataSet** service shall **not be de-**
 4 **leted** as long as they are referenced by any control class (for example, **URCB** or **GoCB**).

5 A non-persistent **DATA-SET** may be accessed using the services **GetDataSetValue**, **Set-**
 6 **DataSetValue**, and **GetDataSetDirectory**, and shall be referenced only by **URCB** and
 7 **USVCB**.

8 NOTE 2 Local reconfiguration of members of a **DATA-SET** may cause critical misoperations. To prevent unin-
 9 tended changes in the **DATA-SET** configuration, special measures have to be taken in a system (the measures are
 10 outside the scope of this part of IEC 61850).

11 11.2 DATA-SET class definition

12 11.2.1 DATA-SET class syntax

13 The **DATA-SET** shall have the structure as defined in Table 21.

14 **Table 21 – DATA-SET (DS) class definition**

DATA-SET class		
Attribute name	Attribute type	Value/value range/explanation
DSName	ObjectName	Instance name of an instance of DATA-SET
DSRef	ObjectReference	Path-name of an instance of DATA-SET
DSMemberRef [1..n]	(*)	(*) Functional constrained data (FCD) or functional constrained data attribute (FCDA)
Services GetDataSetValue SetDataSetValue CreateDataSet DeleteDataSet GetDataSetDirectory		

15
 16

17 11.2.2 DATA-SET class attributes

18 11.2.2.1 DSName

19 The attribute **DSName** shall unambiguously identify **DATA-SET** within the scope of a
 20 **LOGICAL-NODE** or within a two-party-application-association.

21 11.2.2.2 DSRef

22 The attribute **DSRef** shall be the unique path-name of an instance of DATA-SET.

23 The ObjectReference **DSRef** shall be one of the following two options.

LDName/LNName.DataSetName	To reference a persistent instance of DATA-SET
@DataSetName	To reference a non-persistent instance of DATA-SET

24

25 11.2.2.3 DSMemberRef [1..n] – data set member reference

26 The attribute **DSMemberRef** shall define the functional constrained data (**FCD**) or functional
 27 constrained data attribute (**FCDA**) of **DATA**.

1 An IED which claims to support dynamic creation of datasets (CreateDataSet) shall be able to
 2 receive (as a server), ~~send (as a client)~~ and process (as a server ~~or as a client~~) any valid FCD
 3 or FCDA definition contained in the CreateDataSet request. An IED which claims to support
 4 configuration of datasets (via SCL) shall be able to process (as a server or as a client) any
 5 valid FCD or FCDA definition contained in the corresponding SCL file.

6 The value of a member of a **DATA-SET** retrieved, set, reported, or logged shall be determined
 7 by the functional constrained **DATA (FCD)** or functional constrained data attribute (**FCDA**).

8 NOTE A **DATA-SET** does not contain **DATA**. A **DATA-SET** contains references, the functional constrained data
 9 (**FCD**) or functional constrained data attribute (**FCDA**). A **DATA-SET** may contain references to functional con-
 10 strained **DATA (FCD)** or functional constrained data attribute (**FCDA**) contained in different **LOGICAL-NODES**.

11 11.3 DATA-SET class services

12 11.3.1 Overview

13 For **DATA-SET** the following services are defined.

Service	Description
GetDataSetValue	Retrieve all value of DATA referenced by the members of the DATA-SET
SetDataSetValue	Write all value of DATA referenced by the members of the DATA-SET
CreateDataSet	Create a DATA-SET by providing the FCD (FCDA) references or that form the DATA-SET
DeleteDataSet	Delete a DATA-SET
GetDataSetDirectory	Retrieve FCD references of all members referenced in the DATA-SET

14

15 11.3.2 GetDataSetValues

16 11.3.2.1 GetDataSetValues parameter table

17 The client shall use the **GetDataSetValues** service to retrieve the values of all referenced
 18 **DataAttributes** made visible and thus accessible to the requesting client by the referenced
 19 **DATA-SET**.

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

Parameter name
Request
DataSetReference
Response+
DataSetReference
DataAttributeValue [1..n]
Response-
ServiceError

22

23 11.3.2.2 Request

24 DataSetReference – data set ObjectReference

25 The parameter **DataSetReference** shall define the **ObjectReference** of the **DATA-SET**. The
 26 **ObjectReference DataSetReference** shall be one of the following two options.

27 – **LDName/LNName.DataSetName** to reference a **persistent DATA-SET**, or

1 – **@DataSetName** to reference a **non-persistent DATA-SET**.

2 **11.3.2.3 Response+**

3 **DataAttributeValue [1..n]**

4 The parameter **DataAttributeValue** shall contain values of a member of the **DATA-SET**. The
5 value of the **DataAttributes** of the **DATA** may be simple or complex depending on the defini-
6 tion of the **DATA**. For complex **DataAttrTypes** the values of all **DataAttributes** of all nesting
7 levels shall be returned.

8 Each element of the list shall either contain the value of the **DataAttribute** at the time of ac-
9 cess, or a reason for an access error.

10 NOTE The syntax of the **DataAttributeValue** is defined in an SCSM.

11 **11.3.2.4 Response–**

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

14 **11.3.3 SetDataSetValues**

15 **11.3.3.1 SetDataSetValues parameter table**

16 The client shall use the **SetDataSetValues** service to set the values of all **DataAttributes**
17 made visible and thus accessible to the requesting client by the referenced **DATA-SET**.

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

Parameter name
Request
DataSetReference
DataAttributeValue [1..n]
Response+
Result
Response–
ServiceError

20

21 **11.3.3.2 Request**

22 **11.3.3.2.1 DataSetReference – data set ObjectReference**

23 The parameter **DataSetReference** shall define the ObjectReference of a **DATA-SET**. The Ob-
24 jectReference DataSetReference shall be one of the following two options:

- 25 – **LDName/LNName.DataSetName** to reference a **persistent DATA-SET**, or
- 26 – **@DataSetName** to reference a **non-persistent DATA-SET**.

27 **11.3.3.2.2 DataAttributeValue [1..n]**

28 The parameter **DataAttributeValue** shall contain a value of a member of the **DATA-Set**. The
29 value of the **DataAttribute** of the **DATA** may be simple or complex depending on the defini-

1 tion of the **DATA**. For complex **DataAttrTypes** the values of all **DataAttributes** of all nesting
2 levels shall be contained.

3 NOTE The syntax of the **DataAttributeValue** is defined in an SCSM.

4 **11.3.3.3 Response+**

5 The parameter **Response+** shall indicate that the service request succeeded.

6 NOTE The action to be taken by an application receiving the values for the instances of DataAttributes to be set is
7 outside the scope of this service definition.

8 A successful result shall return the following parameter.

9 **Result**

10 The parameter **Result** shall return a list, specified in the order of the **ObjectReferences** of
11 the **DATA** that are referenced in the **DATA-SET**. This list shall indicate, for each **DATA**, either
12 a confirmation that the service **SetDataSetValue** to the referenced instance succeeded or a
13 reason why the service **SetDataSetValue** to the referenced **DATA** failed.

14 **11.3.3.4 Response–**

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

17 **11.3.4 CreateDataSet**

18 **11.3.4.1 CreateDataSet parameter table**

19 The client shall use the **CreateDataSet** service to request the server to create a **DATA-SET**
20 with a list of members defined with the functional constrained data (**FCD**) or functional con-
21 strained data attribute (**FCDA**) made visible and thus accessible to the requesting client.

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

Parameter name
Request
DataSetReference
DSMemberRef [1..n]
Response+
Response–
ServiceError

24

25 **11.3.4.2 Request**

26 **11.3.4.2.1 DataSetReference – data set ObjectReference**

27 The parameter **DataSetReference** shall define the ObjectReference of **DATA-SET** that is to
28 be created. The ObjectReference DataSetReference shall be one of the following two options:

- 29 – **LDName/LNName.DataSetName** to create a **persistent DATA-SET**, or
- 30 – **@DataSetName** to create a **non-persistent DATA-SET**.

1 11.3.4.2.2 DSMemberRef [1..n] – data set member ObjectReference

2 The parameter **DSMemberRef** shall define the functional constrained data (**FCD**) or functional
3 constrained data attribute (**FCDA**) of a **DATA**.

4 11.3.4.3 Response+

5 The parameter **Response+** shall indicate that the service request succeeded. If one of the
6 referenced functional constrained data (**FCD**) are not available to that client then the service
7 shall fail.

8 11.3.4.4 Response–

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

11 11.3.5 DeleteDataSet

12 11.3.5.1 DeleteDataSet parameter table

13 The client shall use the **DeleteDataSet** service to request the server to delete a **DATA-SET**
14 made visible and thus accessible to the requesting client.

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

Parameter name
Request
DataSetReference
Response+
Response–
ServiceError

17

18 11.3.5.2 Request

19 11.3.5.2.1 DataSetReference – data set ObjectReference

20 The parameter **DataSetReference** shall define the **ObjectReference** of a **DATA-SET** that
21 shall be deleted. The **ObjectReference DataSetReference** shall be one of the following two
22 options.

- 23 – **LDName/LNName.DataSetName** to delete a dynamically created **persistent DATA-**
24 **SET**, or
- 25 – **@DataSetName** to delete a **non-persistent DATA-SET**.

26 11.3.5.3 Response+

27 The parameter **Response+** shall indicate that the service request succeeded.

28 11.3.5.4 Response–

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

1 11.3.6 GetDataSetDirectory

2 11.3.6.1 GetDataSetDirectory parameter table

3 The client shall use the **GetDataSetDirectory** service to retrieve the list of the **Object-**
 4 **References** of all data set members referenced by the **DATA-SET** made visible and thus ac-
 5 cessible to the requesting client.

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

Parameter name
Request
DataSetReference
Response+
DSMemberRef [1..n]
Response–
ServiceError

8

9 11.3.6.2 Request

10 DataSetReference – data set ObjectReference

11 The parameter **DataSetReference** shall define the **ObjectReference** of the **DATA-SET**. The
 12 **ObjectReference DataSetReference** shall be one of the following two options:

- 13 – **LDName/LNName.DataSetName** to reference a **persistent DATA-SET**, or
- 14 – **@DataSetName** to reference a **non-persistent DATA-SET**.

15 11.3.6.3 Response+

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

18 DSMemberRef [1..n] – data set member ObjectReference

19 The parameter **DSMemberRef** shall contain the **ObjectReferences** of the members of the
 20 **DATA-SET**.

21 NOTE The syntax of the **DSMemberRef** is defined in an SCSM.

22 11.3.6.4 Response–

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

25 12 Substitution model

26 The substitution model provides the substitution of values of **DataAttributes** whose functional
 27 constraint equals **MX** (for analogue values) or **ST** (for status values). Basically, substitution ap-
 28 plies to **DataAttributes** with FC (= **MX** and **ST**) and to the associated quality attribute. When
 29 substitution is enabled for a specific **DataAttribute**, the **DATA** shall provide the substituted
 30 values instead of the process value to the clients.

1 NOTE 1 Substituting values is part of the normal operation of a substation and has nothing to do with system or
 2 device tests. Tests are performed by setting a logical device into the test mode or setting the data attribute of a
 3 controllable data object (=Test) by a control services to TRUE.

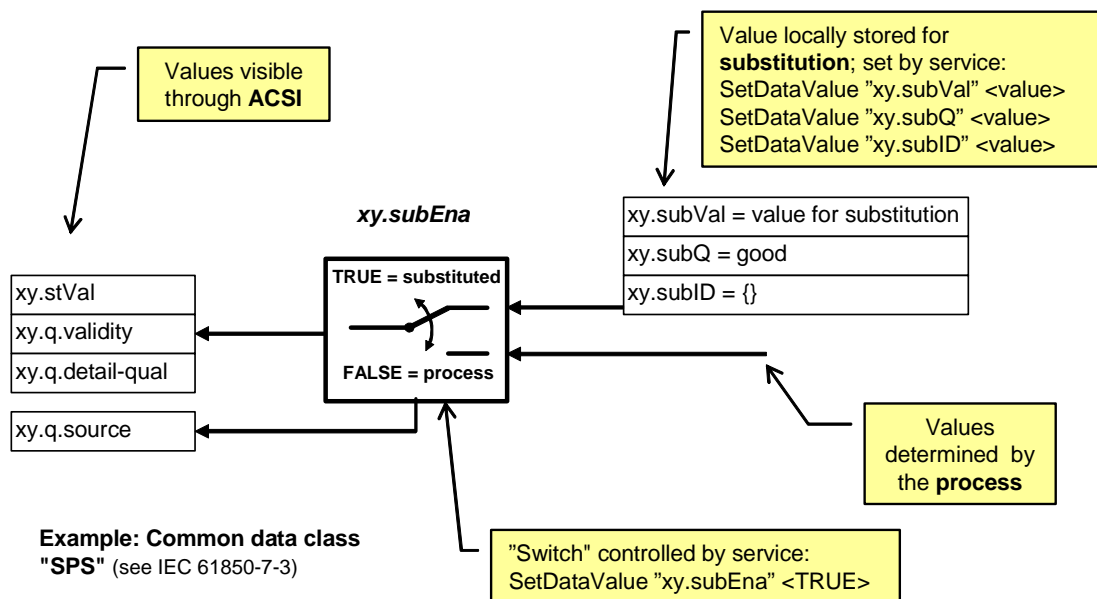
4 In the typical use case for substitution, an operator on the client side enters manually a value
 5 for a **DataAttribute** located in a specific device. The client sets the **DataAttribute** to the
 6 value entered. If a client accesses the value of that **DataAttribute** (for example, using a **Get-**
 7 **dataValue** service or subscribing to a report) the client shall receive the manual entered (sub-
 8 stituted) value instead of the value determined by the process.

9 The substitution model relies on four specific **DataAttributes** defined in IEC 61850-7-3.

- 10 - **subEna** (enable substitution): The current process value shall be replaced by the value
 11 provided in the **DataAttribute subVal**.
- 12 - **subVal**, **subMag**, and **subCMag** (values for substitution of process values): The current
 13 process value shall be replaced by the value provided by the **DataAttribute subVal**,
 14 **subMag** and **subCMag** respectively.
- 15 - **subQ** (value for substitution of quality): The current process value shall be replaced by
 16 the value provided by the **DataAttribute subQ**.
- 17 - **subID** (value to indicate the initiator of the substitution).

18 The detailed definition of these **DataAttributes** (defined in IEC 61850-7-3) shall be followed in
 19 conjunction with this clause.

20 The concept of substitution is shown in Figure 17. Usually, input from the process or the result
 21 of the calculation from a function provides the value of a **DataAttribute** (in that case, the
 22 source is called "process"). In case of substitution, the value of a **DataAttribute** may be pro-
 23 vided by an operator making use of a client. This selection of the source of the value (substitu-
 24 tion value or process value) shall be controlled by the service **SetDataValue** ("**xy.subEna**"
 25 <TRUE>) to substitute or **SetDataValue** ("**xy.subEna**" <FALSE>) to unsubstitute. The service
 26 **SetDataValue** ("**xy.subVal**" <value for substitution>) shall be used to set the substituted value.
 27 There may be cases, where a local automatic function disables substitution, for example, if
 28 blocking of information exchange is disabled or communication is no longer interrupted.



29
 30

IEC 411/03

31 **Figure 17 – Principles of substitution**

1 The client shall set first the value to be substituted (**xy.subVal**, **xy.subQ**, **xy.subID**) and then
2 enable the substitution by setting the attribute **xy.subEna** to TRUE.

3 NOTE 2 In an SCSM mapping it is recommended to use two SetDataValue services: the first to set the value used
4 for substitution and the second to enable the substitution.

5 After subEna is set to True it shall be possible to set new values (over-write the current values)
6 for subVal, subMag, subCMag, subQ, and subID. The updated values shall be used to update
7 the corresponding process values (stVal, ...).

8 **DataAttributes** that provide the possibility of substitution shall have a functional constraint
9 value of **SV** (substitutable value).

10 In case the association over which the substitution has been enabled fails, the substituted
11 value shall remain unchanged. Changes shall be initiated by a service or by local means in the
12 server device.

13 If the client has no direct access to the server responsible for the data acquisition (for example,
14 in a hierarchical system, with a gateway in between, where the client needs to access a proxy),
15 it shall be a local issue of the proxy how to handle the substitution.

1 **13 SETTING-GROUP-CONTROL-BLOCK class model**

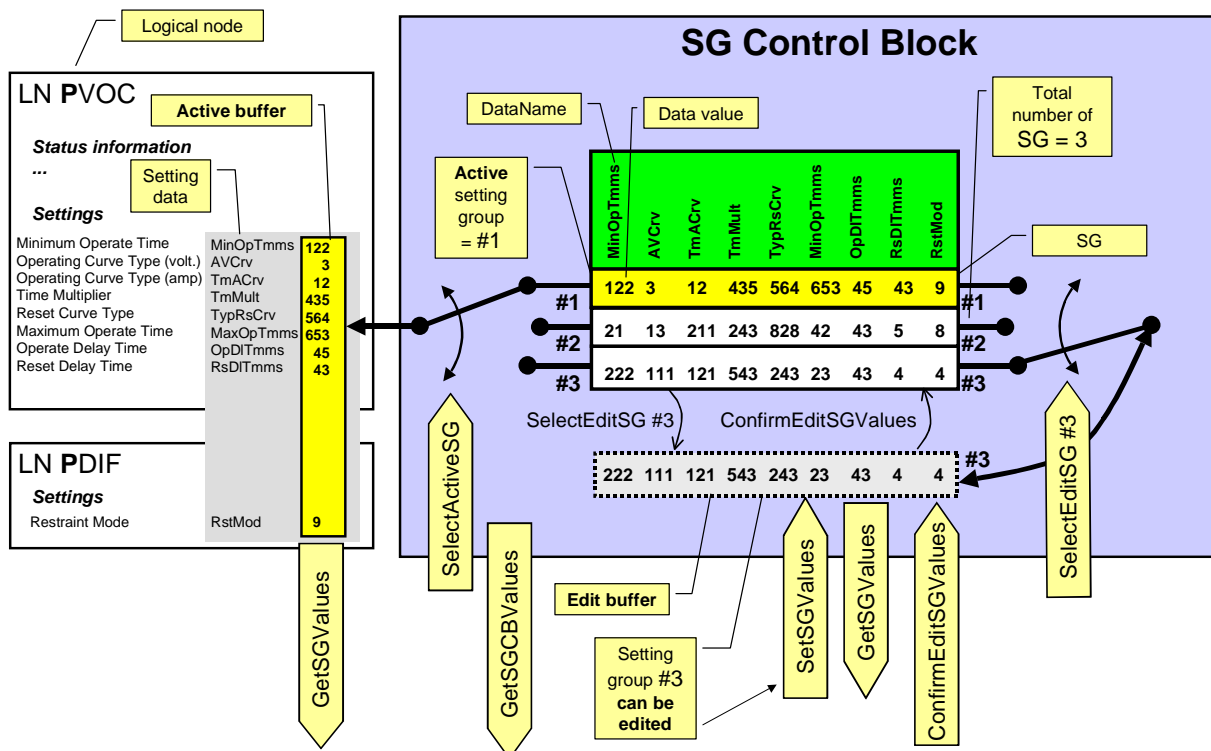
2 **13.1 General**

3 An instance of a **DATA** usually has one value. The **SETTING-GROUP-CONTROL-BLOCK**
 4 (**SGCB**) model allows for an instance to have several values that can be used one at a time.
 5 The **SGCB** provides mechanisms to switch between several values of one or more **DATA**. Val-
 6 ues that belong together build the setting group (**SG**).

7 NOTE A logical node zero (LLN0) may have one **SETTING-GROUP-CONTROL-BLOCK**. Many setting **DATA** are
 8 defined in IEC 61850-7-4.

9 The **SGCB** model provides services to handle different values for one or more **DATA**. The **SG**
 10 whose values are currently used by the **DATA** of a **LOGICAL-NODE** shall be in the state “ac-
 11 tive”. The **SG** that can be edited shall be in the state “edit”.

12 The **SGCB** model is depicted in the example in Figure 18. The **LOGICAL-NODE** “**PVOC**” (volt-
 13 age controlled/dependent time overcurrent according to IEC 61850-7-4) comprises eight **DATA**
 14 for settings (**LN PDIF** has one **DATA** for settings) – (**MinOpTmms**, ..., **RstrMode**). The **SGCB**
 15 “**SG Control**” provides three **SGs** (#1, #2, and #3) each with independent values for the nine
 16 **DATA**. Each **SG** contains nine values (one for each of: **MinOpTmms**, ..., **RstrMode**). The
 17 members of the active **SG** are referenced by the **ObjectReferences** of the **DATA** with func-
 18 tional constraint **SG**. The members of the **SG** in the “edit buffer” are referenced by the **Objec-**
 19 **tReferences** of the **DATA** with functional constraint **SE**.



20
21

IEC 412/03

22 **Figure 18 – Basic model of the settings model**

1 The values of the **DATA** of the **LOGICAL-NODE PVOC** are derived from the values of one of
 2 the **SGs**. This is accomplished by the multiplexer on the left. The service **SelectActiveSG** de-
 3 termines which values (of **SG #1**, **#2**, or **#3**) shall be copied to the “active buffer” and be used
 4 by **PVOC**. In the example the **SG #1** has been set to be in the active state.

5 A **SG** contains values for **DATA** that are contained in several **LOGICAL-NODEs**. The **SGs** in
 6 the example provide values for **DATA** in two **LOGICAL-NODEs (PDIF and PVOC)**.

7 The values of **SG #3** can be edited (the **SelectEditSG** switched the right multiplexer to **#3**);
 8 the values of this **SG** (now in the edit buffer) can be set and get (**SetSGValue** and
 9 **GetSGValue**). After values have been set in the edit buffer (values of **SG #3**), the cli-
 10 ent shall confirm that the new values (stored in the edit buffer) shall be taken over by
 11 the selected **SG (SG #3)**.

12 The attributes of the **SGCB** can be retrieved (**GetSGCBValue**).

13 The **DATA** contained in the **SG** can be accessed directly with **GetSGValue**.

14 13.2 SGCB class definition

15 13.2.1 SGCB class syntax

16 The **SGCB** shall have the structure defined in Table 22.

17 Clients should use the existence of a **SGCB** to determine if the **LOGICAL-DEVICE** contains SGs.

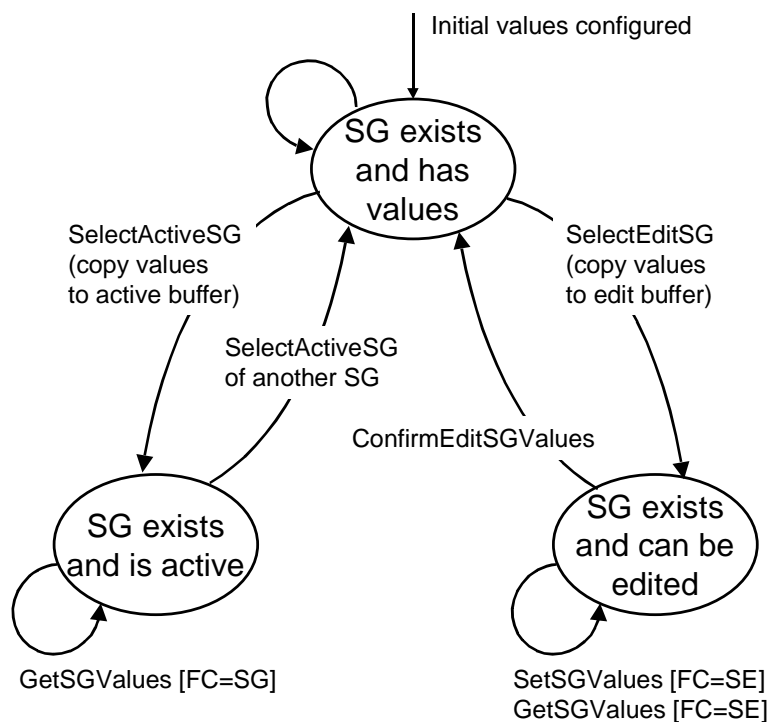
18 **Table 22 – SGCB class definition**

SGCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
SGCBName	ObjectName	-	-	Instance name of an instance of SGCB
SGCBRef	ObjectReference	-	-	Path-name of an instance of SGCB
NumOfSG	INT8U	SP	-	n = NumOfSG
ActSG	INT8U	SP	dchg	Allowable range: 1 ... n
EditSG	INT8U	SP	dchg	Allowable range: 0 ... n
CnfEdit	BOOLEAN	SP	dchg	
LActTm	TimeStamp	SP	dchg	
Services				
SelectActiveSG				
SelectEditSG				
SetSGValue				
ConfirmEditSGValue				
GetSGValue				
GetSGCBValue				

19

20 Values of the attributes of the instances of **SGCB** shall be configured.

21 The setting group shall behave as shown in Figure 19.



IEC 413/03

Figure 19 – Setting group state machine

The state changes shall be issued as defined with the corresponding attributes of the **SGCB** and the corresponding services of the **SGCB**.

13.2.2 SGCB class attributes

13.2.2.1 SGCBName – setting group control name

The attribute **SGCBName** shall be **SGCB** within the scope of a **LLNO**.

13.2.2.2 SGCBRef – setting group control ObjectReference

The attribute **SGCBRef** shall be the unique path-name of an **SGCB**.

The ObjectReference **SGCBRef** shall be:

LDName/LLNO.SGCB

NOTE **SGCB** is the standardized instance name of the **SGCB**.

13.2.2.3 NumOfSG – number of setting groups

The attribute **NumOfSG** shall identify the total number of **SGs** that are available in a **LOGICAL-DEVICE**.

If there are any **DataAttributes** with functional constraint **SG** in a **LOGICAL-DEVICE** then a single **SGCB** shall be present in the **LOGICAL-DEVICE**.

The attribute **NumOfSG** shall not be settable. The value of **NumOfSG** is a local matter.

1 13.2.2.4 ActSG – active setting group

2 The attribute **ActSG** shall identify the values of the **SG** that are in the active buffer. The attrib-
3 ute **ActSG** shall define the **SG** whose values shall be used by the respective **LOGICAL-NODE**
4 to performing its function. The values the **DataAttribute** of the active **SG** can retrieved by the
5 service **GetSGValue**.

6 13.2.2.5 EditSG – edit setting group

7 The attribute **EditSG** shall identify the values of the **SG** in the edit buffer. The values of the edit
8 buffer can be set and retrieved by the services **SetSGValue** and **GetSGValue**. The original
9 values in the **SG** shall be unchanged until the client has confirmed to overwrite the values with
10 those values stored in the edit buffer (**ConfirmEditSGValue**).

11 If the value of **EditSG** is (= 0) then the use of services **SetSGValue** (with **FC=SE**) and
12 **GetSGValue** shall cause a **Response–**.

13 13.2.2.6 CnfEdit – confirm editing

14 The attribute **CnfEdit** shall be used to confirm the editing process.

15 13.2.2.7 LActTm – last activation time

16 The attribute **LActTm** shall identify the time when the last service **SelectActiveSG** has been
17 processed.

18 13.3 SGCB class services

19 13.3.1 Overview

20 For **SGCB** the following services are defined.

Service	Description
SelectActiveSG	Select which SG shall become the active SG
SelectEditSG	Select which SG shall become the SG that can be edited after selecting
SetSGValue	Write value to the SG which has been selected for editing
ConfirmEditSGValue	Confirm that the new value to the SG which has been selected for editing become the value of the SG
GetSGValue	Read value from the SG which has been selected for editing (FC = SE) or of the active SG (FC = SG)
GetSGCBValue	Read all attribute value of the SGCB

21

22 13.3.2 SelectActiveSG

23 13.3.2.1 SelectActiveSG parameter table

24 A client shall use the **SelectActiveSG** service to load the values of the specified **SG** into the
25 active buffer.

Parameter name
Request
SGCBReference
SettingGroupNumber
Response+
Response–
ServiceError

1 **13.3.2.2 Request**

2 **13.3.2.2.1 SGCBReference**

3 The parameter **SGCBReference** shall contain the ObjectReference **LDName/LLN0.SGCB**.

4 **13.3.2.2.2 SettingGroupNumber**

5 The parameter **SettingGroupNumber** shall define the number **ActSG** of the **SG** (between 1
6 and **NumOfSG**) that shall be used to determine the new values of **DATA** of the respective
7 **LOGICAL-NODEs**.

8 The values of all instances of the setting **DATA** of all **LOGICAL-NODEs** (that get their setting
9 values from the setting group specified in the service request) shall be over-written with the
10 new values of the data of the setting group referenced in the service request.

11 **13.3.2.3 Response+**

12 The parameter **Response+** shall indicate that the service request succeeded.

13 **13.3.2.4 Response–**

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

16 **13.3.3 SelectEditSG**

17 **13.3.3.1 SelectEditSG parameter table**

18 A client shall use the **SelectEditSG** service to set the **EditSG** value of the referenced **SGCB**
19 made visible and thus accessible to the requesting client by the referenced **LLNO**.

20 It is the client's responsibility to check the attributes of a **SGCB** before it continues editing (con-
21 firming) the setting group in the edit buffer after an association was down. After loss of an as-
22 sociation the **SelectEditSG** service shall be re-issued to copy the values of the selected **SG** to
23 the edit buffer.

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

Parameter name
Request
SGCBReference
SettingGroupNumber
Response+
Response–
ServiceError

26

27 **13.3.3.2 Request**

28 **13.3.3.2.1 SGCBReference**

29 The parameter **SGCBReference** shall contain the **ObjectReference** of the **SGCB**.

30 The **ObjectReference SGCBReference** shall be:

1 **LDName/LLNO.SGCB**

2 **13.3.3.2.2 SettingGroupNumber**

3 The parameter **SettingGroupNumber** shall define the number **EditSG** of the **SG** (between 1
4 and **NumOfSG**) that shall be used to set values (**SetSGValue**), confirm value (**Con-**
5 **firmEditSGValue**), and retrieve value (**GetSGValue**) of the specified **SG**.

6 **13.3.3.3 Response+**

7 The parameter **Response+** shall indicate that the service request succeeded.

8 **13.3.3.4 Response–**

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

11 **13.3.4 SetSGValue**

12 **13.3.4.1 SetSGValue parameter table**

13 A client shall use the **SetSGValue** service to set the value of the **DATA** of the **SG** identified by
14 the value of the attribute **EditSG** of the **SGCB** made visible and thus accessible to the request-
15 ing client by the referenced **LLNO**.

16 Setting new value shall become effective only after the client has confirmed the value by issu-
17 ing the service **ConfirmEditSGValue**.

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

Parameter name
Request
Reference
DataAttributeValue [1..n]
Response+
Response–
ServiceError

20

21 **13.3.4.2 Request**

22 **13.3.4.2.1 Reference**

23 The parameter **Reference** shall define the **functional constrained data (FCD)** or **func-**
24 **tional constrained data attributes (FCDA)** of the **DATA** whose **DataAttribute** value is to
25 be written. The **Reference** shall be **FCD** or **FCDA**.

26 The **FunctionalConstraint** value of the **FCD** or **FCDA** shall be **SE**.

27 **13.3.4.2.2 DataAttributeValue [1..n]**

28 The parameter **DataAttributeValue** shall contain

- 29 – the values of all **DataAttributes** of a **DATA** referenced by **FCD**; or
- 30 – the value of a **DataAttribute** referenced by **FCDA**

1 of the **SG** identified by the value of the attribute **EditSG** of the **SGCB**.

2 NOTE The syntax of the **DataAttributeValue** is defined in an SCSM.

3 13.3.4.3 Response+

4 The parameter **Response+** shall indicate that the service request succeeded.

5 13.3.4.4 Response–

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

8 13.3.5 ConfirmEditSGValue

9 13.3.5.1 ConfirmEditSGValue parameter table

10 A client shall use the **ConfirmEditSGValue** service to confirm that the value of the **SG** (identified by the attribute **EditSG**) set with the service **SetSGValue** shall overwrite the old value of the **SG** of the **SGCB** made visible and thus accessible to the requesting client by the referenced **LLNO**.

14 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
SGCBReference
Response+
Result
Response–
ServiceError

16

17 13.3.5.2 Request

18 SGCBReference

19 The parameter **SGCBReference** shall contain the **ObjectReference LDName/LLNO.SGCB**.

20 13.3.5.3 Response+

21 The parameter **Response+** shall indicate that the service request succeeded.

22 13.3.5.4 Response–

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

25 13.3.6 GetSGValue

26 13.3.6.1 GetSGValue parameter table

27 A client shall use the **GetSGValue** service to retrieve the value of **DATA** of **SGs** made visible and thus accessible to the requesting client by the referenced **LLNO**.

28

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

Parameter name
Request
Reference
Response+
DataAttributeValue [1..n]
Response–
ServiceError

3

4 13.3.6.2 Request

5 Reference

6 The parameter **Reference** shall define the **functional constrained data (FCD)** or **func-**
7 **tional constrained data attributes (FCDA)** of the **DATA** whose **DataAttribute** value are to
8 be retrieved. The **Reference** shall be **FCD** or **FCDA**.

9 The **FC** value of the **FCD** or **FCDA** shall be

- 10 – **SE** to retrieve the value of the **SG** in the edit buffer; and
- 11 – **SG** to retrieve the value of the active **SG**.

12 13.3.6.3 Response+

13 DataAttributeValue [1..n]

14 The parameter **DataAttributeValue** shall contain

- 15 – the value of all **DataAttributes** of a **DATA** referenced by **FCD**; or
- 16 – the value of a **DataAttribute** referenced by **FCDA**.

17 The **FC** value of the **FCD** or **FCDA** shall be **SE** or **SG** respectively.

18 NOTE The syntax of the **DataAttributeValue** is defined in an SCSM.

19 13.3.6.4 Response–

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

22 13.3.7 GetSGCBValue

23 13.3.7.1 GetSGCBValue parameter table

24 A client shall use the **GetSGCBValue** service to retrieve the list of attribute values of the refer-
25 enced **SGCB** made visible and thus accessible to the requesting client by the referenced **LLNO**.

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

Parameter name
Request
SGCBReference
FunctionalConstraint
Response+
NumberOfSettingGroup
ActiveSettingGroup
EditSettingGroup
LastActivateTime
Response–
ServiceError

3

4 **13.3.7.2 Request**

5 **13.3.7.2.1 SGCBReference**

6 The parameter **SGCBReference** shall contain the **ObjectReference LDName/LLN0.SGCB**.

7 **13.3.7.2.2 FunctionalConstraint**

8 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
9 rameter to identify the functional constraint of the respective attribute of **SGCB** contained in the
10 **LLN0**. The value shall be **SP**.

11 **13.3.7.3 Response+**

12 **13.3.7.3.1 NumberOfSettingGroup – number of setting group controls**

13 The parameter **NumberOfSettingGroup** shall define the total number of the **SG** of the attrib-
14 ute **NumOfSG** of the referenced **SGCB**.

15 **13.3.7.3.2 ActiveSettingGroup – active setting group**

16 The parameter **ActiveSettingGroup** shall define the number of the **SGs** of the attribute **Ac-**
17 **tiveSG** from which the current active **SG** values shall be derived.

18 **13.3.7.3.3 EditSettingGroup – edit setting group**

19 The parameter **EditSettingGroup** shall define the number of the **SG** of the attribute **EditSG**
20 whose values can be set and retrieved.

21 **13.3.7.3.4 LastActivateTime – last time of activation of a setting group**

22 The parameter **LastActivateTime** shall define the time of the last activation of the attribute
23 **LActTm**.

24 **13.3.7.4 Response–**

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

27

1 **14 REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK class models**

2 **14.1 Overview**

3 Reporting and logging meets a number of crucial requirements for event-driven information ex-
4 change. The data transfer models described in this clause provide mechanisms for transferring
5 data values caused by well-defined conditions from a logical node to one client or storing the
6 data in a server's log for future querying.

7 In contrast to high bandwidth and time-consuming fast reading (polling) devices for extraordi-
8 nary event occurrences, the reporting provides immediate transmission of events. Reporting is
9 controlled by constraints.

10 The main characteristics of reporting and logging are:

- 11 – timely reports serve as an indication to clients under real-time constraints (optionally keep-
12 ing sequence-of-events to the client),
- 13 – logging of events for later retrieval (sequence-of-events stored in server),
- 14 – the impact on network bandwidth is minimized,
- 15 – sending reports only when required (controlled by several attributes),
- 16 – low-frequency integrity scan and client-initiated general interrogation.

17 Reporting provides mechanisms to report packed values of instances of **DATA** immediately or
18 after some buffer time. The logging model provides mechanisms to store events in the log in
19 sequence. A client may query a range of log entries at any time.

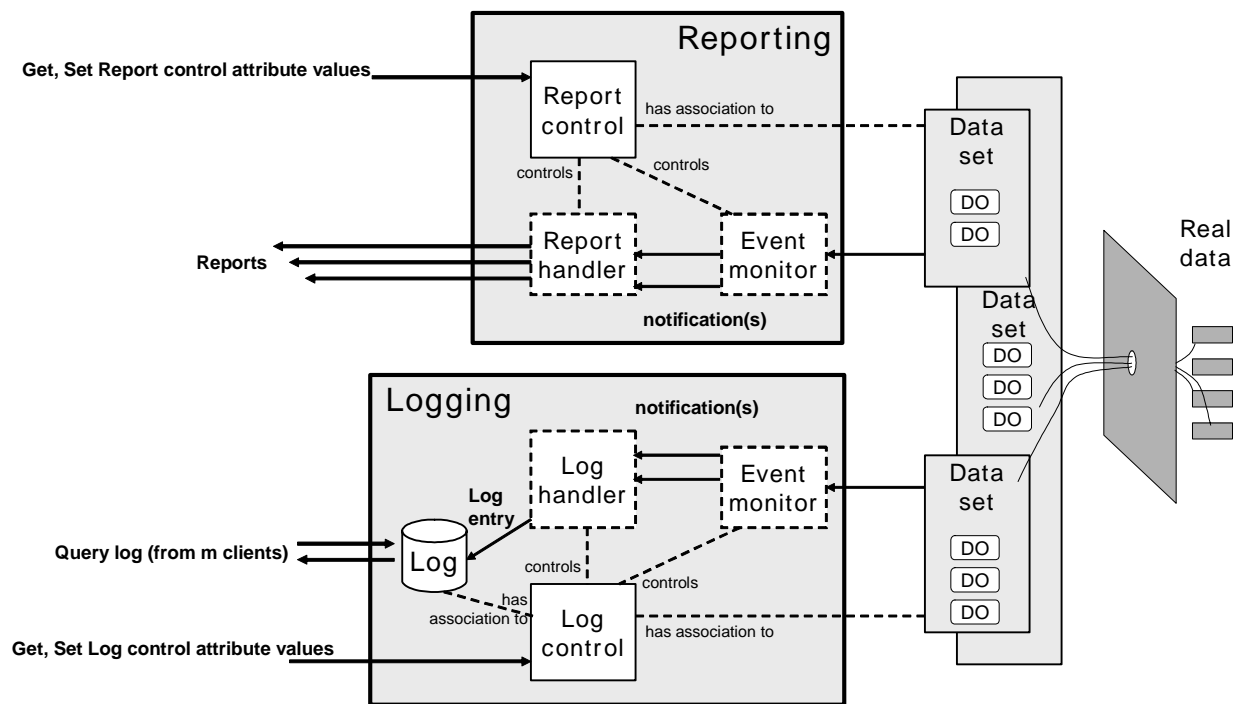
20 Reporting and logging as well as the basic services of the data model provide flexible data re-
21 trieval schemas, for example:

- 22 – change-of-state notification of clients: immediate reports,
- 23 – sequence-of-events: keeping reports in sequence or storing and querying sequences of log
24 entries,
- 25 – polling data at any time: GetDataValues and GetDataSetValues

26 NOTE 1 Subclause 14.3.5.3.4 provides special services for event distribution (generic substation event model,
27 **GSE**). Reporting and **GSE** have totally different qualities of services and behaviour. Reporting is connection-oriented
28 (**GSE** uses multicast), reporting transmits data once (**GSE** transmits and retransmits data with heartbeat).
29 IEC 61850-7-1 provides a comparison of the models.

30 NOTE 2 Clause 16 specifies special services for communication of measured values of, for example, voltage
31 transformer (VT) and current transformer (CT) under crucial time constraints.

32 The principle building blocks and services for reporting and logging are depicted in Figure 20.



IEC 414/03

1
2
3 **Figure 20 – Basic building blocks for reporting and logging**
4

5 The reporting model is composed of three building blocks. The logging model has four building
6 blocks. Classes are defined for the report control, the log control and the log.

7 NOTE 3 The handler and monitor are introduced here for conceptual reasons only.

8 The **DATA-SET** (referencing **DATA**) represent the real data values. The real data values are
9 conceptually monitored by the event monitors. An event monitor determines, on the basis of the
10 state of the real data and the attributes of the control class, when to generate a notification to
11 the appropriate handlers (e.g. Log or Report Handler). This notification includes the data val-
12 ues and reasons for data inclusion.

13 Note: The contents of the event notification are determined by a combination of I/O scan and
14 event monitoring. These are conceptually two asynchronous processes. Therefore, the number
15 of data values included within a single notification is a local issue.

16 The number of data values, within a notification, is a local implementation issue. The report
17 handler assigns EntryID(s) and TimeOfEntry(s) to the values contained within a set of notifica-
18 tions. The number of notifications combined into a single EntryID is determined by the RCB
19 control parameters (e.g. BufTim). The value of the EntryID is a local issue but it shall be a
20 unique arbitrary OCTETSTRING whose value is unique within the scope of entries for a spe-
21 cific RCB. The value of the TimeOfEntry shall be the timestamp representing the time at which
22 the report handler received the first notification that is used to form an EntryID. The report
23 handler decides when and how to send a report to the subscribed client. The log handler stores
24 a log entry to the log.

25 A client may initiate a general interrogation at any time to receive all data values of an applica-
26 tion specific set of data. Using this mechanism, clients can synchronize their databases with
27 the current status of a logical node.

1 The **QueryLog** service provides retrieval of a set of selective log entries. Selection criteria are
2 the time range or the range of entryIDs.

3 **14.2 REPORT-CONTROL-BLOCK class model**

4 **14.2.1 Basic concepts**

5 The **REPORT-CONTROL-BLOCK** shall control the procedures that are required for reporting
6 values of **DATA** from one or more **LOGICAL-NODEs** to one client. Instances of report control
7 shall be configured in the server at configuration time.

8 A server shall restrict access to an instance of a report control to one client at a time. That cli-
9 ent exclusively shall “own” that instance and shall receive reports from that instance of report
10 control.

11 There are two classes of report control defined, each with a slightly different behaviour.

12 – **BUFFERED-REPORT-CONTROL-BLOCK (BRCB)** – internal events (caused by trigger op-
13 tions data-change, quality-change, and data-update) issue immediate sending of reports or
14 buffer the events (to some practical limit) for transmission, such that values of **DATA** are
15 not lost due to transport flow control constraints or loss of connection. **BRCB** provides the
16 sequence-of-events (**SOE**) functionality.

17 – **UNBUFFERED-REPORT-CONTROL-BLOCK (URCB)** – internal events (caused by trigger
18 options data-change, quality-change, and data-update) issue immediate sending of reports
19 on a “best efforts” basis. If no association exists, or if the transport data flow is not fast
20 enough to support it, events may be lost.

21 To allow multiple clients to receive the same values of **DATA**, multiple instances of the report
22 control classes shall be made available.

23 Report control block instances are named. These names must be unique within the scope of a
24 Logical-Node. The number of instances, that may be visible to a specific client, is a local im-
25 plementation issue and must be appropriately reflected in the configuration provided (e.g.
26 SCL). Once a report control block is reserved, by a specific client, no other client shall have
27 access rights to set the control block parameters.

28 **Buffered report control blocks** are usually configured to be used by a specific client imple-
29 menting a well-defined functionality, for example, a SCADA master. The client may know the
30 ObjectReference of the **BRCB** by configuration or by the use of a naming convention.

31 **14.2.2 BUFFERED-REPORT-CONTROL-BLOCK (BRCB) class definition**

32 **14.2.2.1 BRCB class Syntax**

33 The **BRCB** class shall have the structure defined in Table 23.

1

Table 23 – BRCB class definition

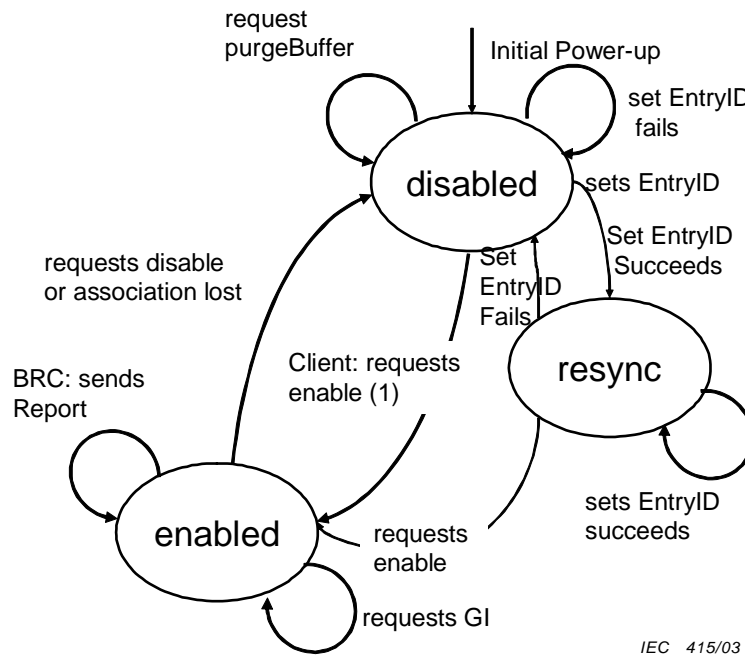
BRCB class					
Attribute name	Attribute type	FC	TrgOp	r/w	Value/value range/explanation
BRCBName	ObjectName	-	-		Instance name of an instance of BRCB
BRCBRef	ObjectReference	-	-		Path-name of an instance of BRCB
Specific to report handler					
RptID	VISIBLE STRING65	BR	-	rw	c1, c2
RptEna	BOOLEAN	BR	dchg	rw	
DatSet	ObjectReference	BR	dchg	rw	c1, c2
ConfRev	INT32U	BR	dchg	r	
OptFlds	PACKED LIST	BR	dchg	rw	c2
sequence-number	BOOLEAN				
report-time-stamp	BOOLEAN				
reason-for-inclusion	BOOLEAN				
data-set-name	BOOLEAN				
data-reference	BOOLEAN				
buffer-overflow	BOOLEAN				
entryID	BOOLEAN				
conf-revision	BOOLEAN				
BufTm	INT32U	BR	dchg	rw	c1, c2
SqNum	INT16U	BR	-	r	
TrgOps	TriggerConditions	BR	dchg	rw	c1, c2
IntgPd	INT32U	BR	dchg	rw	c1, 0.. MAX; 0 implies no integrity report.
GI	BOOLEAN	BR	-	rw	
PurgeBuf	BOOLEAN	BR	-	rw	
EntryID	EntryID	BR	-	rw	c2
TimeOfEntry	EntryTime	BR	-	r	
ResvTms	INT16	BR	-	rw	c3
Services					
Report GetBRCBValues SetBRCBValues					
Notes and Conditions					
Note: An attribute that is marked “r” indicates that the BRCB attribute may be obtained (e.g. read) through the use of the GetBRCBValues service. An attribute that is marked “w” indicates that the BRCB attribute may be set (e.g. written) through the use of the SetBRCBValues service.					
c1: These attributes may only be set when RptEna = FALSE. If a SetBRCBValues service is executed against these attributes and causes a change of value, the implementation shall execute a purge of the buffered events as if PurgeBuf had been set to TRUE.					
c2: These attributes may only be set when RptEna = FALSE. A SetBRCBValues of these parameters, when RptEna=TRUE, shall fail.					
c3: This attribute is optional. If the attribute is not present, then the reservation of the control block shall occur based upon pre-configuration or the client that performs the first SetBRCBValues with RptEna=TRUE.					

2

3

1 The use of a particular instance of a **BRCB** (e.g. which remote clients can actually use the
 2 SetBRCBValues service on a particular **BRCB**) is typically pre-configured locally. The means of
 3 performing this configuration is a local issue. However, it is assumed that one or more cooper-
 4 ating clients (e.g. primary and secondary SCADA systems) may be allowed to have access to a
 5 single **BRCB**. For each set of cooperating clients, for which the server is to provide buffered
 6 reporting service, different **BRCBs** should be provided.

7



8

IEC 415/03

9

Figure 21 – BRCB state machine

10 **disabled:** the **BRCB** is available. No reports shall be issued.

11 Upon power-up, buffering shall be started if the BRCB's DataSet attribute contains a refer-
 12 ence to an existing DataSet. If the DataSet attribute contains a NULL value or a refer-
 13 ence to an invalid DataSet, then no transition shall be allowed from the disabled state.

14 When a client uses SetBRCBValues to set the EntryID attribute value and the set EntryID
 15 does not exist within the queue of entries, a ServiceError of parameter-value-
 16 inappropriate shall be returned.

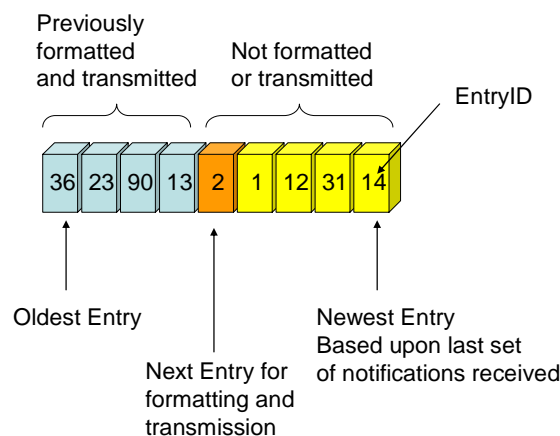
17 If the EntryID value, in the SetBRCBValues, is zero(0), this value is reserved to be used
 18 by the client to resync to the first entry in the queue, a transition from disabled to resync
 19 shall occur.

20 If the set value of the EntryID exists within the queue of entries, a SetBRCBValues re-
 21 sponse+ is returned and the BRCB state shall transition from **disabled** to **resync**.

22 When a client uses SetBRCBValues to set RptEna=TRUE, the state shall transition to the
 23 enabled state.

24 **resync:** the **BRCB** is available. No reports shall be issued.

- 1 When a client uses SETBRCBValues to set an EntryID attribute value and the EntryID
2 value exists within the queue of entries, a SetBRCBValues response+ is returned and the
3 BRCB state shall remain resync.
- 4 If the value of the set EntryID does not exist within the queue of entries, a ServiceError of
5 parameter-value-inappropriate shall be returned and the BRCB state shall transition to
6 **disabled**.
- 7 If the EntryID value, in the SetBRCBValues, is zero(0), this value is reserved to be used
8 by the client to resync to the first entry in the queue, the state shall transition from resync
9 to disabled.
- 10 When an association is lost the state shall transition to **disabled**.
- 11 When a client uses SetBRCBValues to set RptEna=TRUE the state shall transition to **en-**
12 **abled**.
- 13 **enabled**: the **BRCB** shall generate reports for the buffered events and new events as specified
14 in the **BRCB**.
- 15 When an association is lost the state shall transition to **disabled**.
- 16 When a client uses SETBRCBValues to set RptEna=FALSE the state shall transition to
17 disabled and reporting shall cease
- 18 These attributes determine the service procedures of the **Report** service. The impact of the
19 various values shall be as defined in the following attribute definitions.
- 20 Logically, the Report Handler has a queue of entries that are used to sequence the formatting
21 and “transmission” (e.g. queuing to the N-1 layer) reports. The Report Handler, logically, has a
22 pointer to the next entry to be queue for formatting and transmission.



23

24

Figure 22 – General Queue of Entries for Report Handler

25 14.2.2.2 BRCBName – buffered report control name

26 The attribute **BRCBName** shall be the name of the **BRCB** that unambiguously identifies the
27 **BRCB** within a **LOGICAL-NODE**.

28 14.2.2.3 BRCBRef – buffered report control ObjectReference

29 The attribute **BRCBRef** shall be the unique path-name of a **BRCB**.

1 The **ObjectReference BRCBRef** shall be:

LDName/LNName. BRCBName

2
3 **14.2.2.4 RptID – report identifier**

4 The attribute **RptID** shall be the client-specified report identifier of the **BRCB** that has caused
5 the generation of the report. If the report identifier value of the **BRCB** is NULL, then the refer-
6 ence of the **BRCB** shall be reported as the report identifier.

7 NOTE The report identifier field may be used by clients to distinguish between reports from various **BRCBs**. This
8 value is mirrored by the server.

9 **14.2.2.5 RptEna – report enable**

10 The attribute **RptEna** shall be used to control and indicate the current state of the **BRCB**. The
11 state machine for the attribute **RptEna** shall be as depicted in Figure 22.

12 The **BRCB** shall monitor the values of the **DataAttributes** referenced by the **DATA-SET**. In-
13 ternal events as result of the trigger conditions data-change (**dchg**), quality-change (**qchg**),
14 and data-update (**dupd**) shall be buffered (up to a practical limit).

15 The value of RptEna, in the disabled and resync states of Figure 21, shall be FALSE.

16 The client shall configure the **BRCB** and shall then set this attribute to **enabled** (see (1) in
17 Figure 22).

18 While in the state **enabled** no changes of attribute values of the **BRCB** shall be allowed except
19 disabling and activating general-integration.

20 **14.2.2.6 DatSet – Data set reference**

21 The attribute **DatSet** shall specify the **ObjectReference** of the **DATA-SET** being monitored
22 and whose values of the members of the **DATA-SET** (one, a subset, or all) shall be reported.

23 The **DatSet** attribute value shall be included in the report if data-set-name in **OptFlds** of the
24 **BRCB** is set to TRUE otherwise it shall be omitted in the report.

25 A SetReportControlValues of the attribute **DatSet** shall have the same effect as setting **pur-**
26 **geBuf** to TRUE. If the DatSet reference is a valid DataSet or NULL, the SetReportControlVal-
27 ues service shall indicate success. If the DataSet reference is invalid, then the SetReportCon-
28 trolValues service shall indicate failure of badValue and the value of the attribute shall remain
29 unchanged.

30 It is up to local DataSet configuration and client DataSet creation to insure that DataSet mem-
31 bers have TrgOps defined. DataSet members, that have no TrgOps defined, shall only be re-
32 ported due to Integrity and GI reports.

33 **14.2.2.7 ConfRev – configuration revision**

Value is split: See new
table in 7-3.

34 The attribute **ConfRev** shall represent a count of the number of times that the configuration of
35 the **DATA-SET** referenced by **DatSet** has been changed. Changes that shall be counted are:

- 36 – any deletion of a member of the **DATA-SET**;
- 37 – the reordering of members of the **DATA-SET**; and

- 1 – Successful SetxRCBValues of the DatSet attribute where the DatSet attribute value
2 changes.

3 The counter shall be incremented when the configuration changes. At configuration time, the
4 configuration tool will be responsible for incrementing/maintaining the ConfRev value. When
5 configuration changes occur due to SetxRCBValues, the IED shall be responsible for incre-
6 menting the value of ConfRev.

7 The initial value for **ConfRev** is outside the scope of this part of IEC 61850. The value of 0
8 shall be reserved. The value of ConfRev, upon a restart of the IED, is a local issue.

9 Note: There are two possible options for the initializing of the value of ConfRev and the associated configuration:
10 the value is restored to the original value of the base local configuration or the value is retained from the the con-
11 figuration prior to restart. The PICS shall clarify which is implemented.

12 14.2.2.8 OptFlds – optional fields to include in report

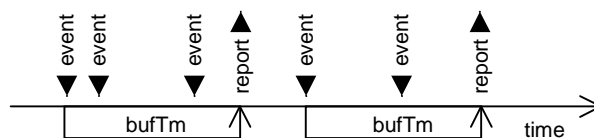
13 The attribute **OptFlds** shall be the client-specified optional fields to be included in the report
14 issued by this **BRCB**. This attribute defines a subset of the optional header fields of the report
15 (see 14.2.3.2.2.1) that shall be included in the report:

- 16 – sequence-number (if TRUE **SqNum** shall be included in the report);
- 17 – report-time-stamp (if TRUE **TimeOfEntry** shall be included in the report);
- 18 – reason-for-inclusion (if TRUE **ReasonCode** shall be included in the report);
- 19 – data-set-name (if TRUE **DatSet** shall be included in the report);
- 20 – data-reference (if TRUE **DataRef** or **DataAttributeReference** shall be included in the re-
21 port);
- 22 – buffer-overflow (if TRUE **BufOvfl** shall be included in the report);
- 23 – entryID (if TRUE **EntryID** shall be included in the report);
- 24 – conf-revision (if TRUE **ConfRev** shall be included in the report).

25 If a **BRCB** does not support one of the above options, then an attempt to set the corresponding
26 bit to TRUE shall cause a negative response of the **SetBRCBValues** service.

27 14.2.2.9 BufTm – buffer time

28 The attribute **BufTm** (see Figure 23) shall specify the time interval in milliseconds for the buff-
29 ering of internal notifications caused by data-change (**dchg**), quality-change (**qchg**), data-
30 update (**dupd**) by the **BRCB** for inclusion into a single report.



IEC 416/03

32 **Figure 23 – Buffer time**

33 Upon receipt of the first set of internal notification of events of the referenced **DATA-SET**, the
34 **BRCB** shall start a timer of the duration buffer time. The number of events within the internal
35 notification is a local issue. When the timer expires, the **BRCB** shall combine all internal notifi-
36 cations that have been received during the time interval into a single report. The next internal
37 notification following the timer expiration shall signal the new start of that timer.

38 The default value of 0 shall be reserved to indicate that the buffer time attribute is not to be
39 used by the **BRCB**. In this case, each internal notification shall cause the **BRCB** to send a sin-

1 gle report. The value shall be settable in 1 ms increments and shall be able to convey up to 1 h
2 of buffer time.

3 NOTE 1 The standard does not require a specific implementation of the monitoring function in a server. The
4 mechanism of how to monitor the application data is outside the scope of this part of IEC 61850. An internal event
5 is understood as an abstract internal indication that, for example, a specific status value has been changed.

6 In the case where a second internal notification of the same member of a **DATA-SET** has oc-
7 curred prior to the expiration of **BufTm**, the **BRCB**

- 8 – shall for status (FC=ST) information behave as if **BufTm** has expired and immediately send
9 the report, restart the timer with value **BufTm** and process the second notification; or
- 10 – may for analogue (FC=MX) information behave as if **BufTm** has expired and immediately
11 transmit the report for transmission, restart the timer with value **BufTm** and process the
12 second notification; or
- 13 – may for analogue information (FC=MX) substitute the current value in the pending re-
14 port with the new one.

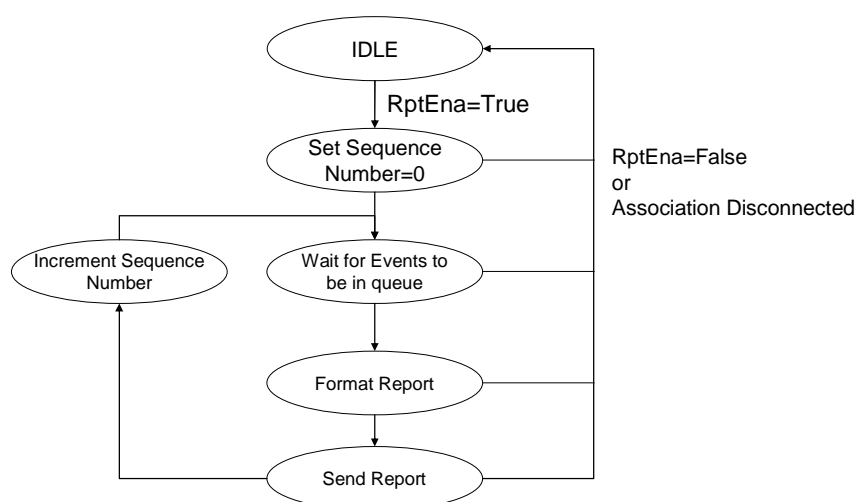
15 NOTE 2 Changes of the same member are communicated in consecutive reports. No reports will be lost because
16 the **BRCB** buffers them.

17 If a **BRCB** does not support buffer time then an attempt to set the **BufTm** attribute to a value
18 greater than zero shall cause a negative response of the **SetReportControlValues** service.

19 A SetReportControlValues of the attribute **BufTm** shall have the same effect as setting **pur-**
20 **geBuf** to TRUE if the attribute value changes.

21 14.2.2.10 SqNum – sequence number

22 The attribute **SqNum** shall specify the sequence number for each **BRCB** that has report enable
23 set to TRUE. This number is to be incremented by the **BRCB** for each report generated and
24 sent. The increment shall occur once the **BRCB** has formatted the report and and requested for
25 transmission.,



26

27 **Figure 24 – State Machine for Sequence Number Generation**

28

29 Figure 24 depicts the logical state machine used in the generation of SqNum. The transition
30 from the IDLE state is caused by the transition of RptEna changing from a value of FALSE to a

1 value of TRUE. Upon that transition, the value of SqNum shall be set to zero(0). As Reports
2 are sent, the value of SqNum shall be incremented. When the maximum value of SqNum is ob-
3 tained, the next value shall be a value of zero(0). .

4 Note: Subsequent writes of RptEna=TRUE, that do not cause the transition from FALSE to
5 TRUE, shall not cause the value of SqNum to be set to zero (0).

6 **14.2.2.11 TrgOps – trigger options**

7 The attribute **TrgOps** shall specify the trigger conditions which shall be monitored by this
8 **BRCB**. The following values are defined:

- 9 – **data-change (dchg)**
- 10 – **quality-change (qchg)**
- 11 – **data-update (dupd)**
- 12 – **integrity**
- 13 – **general-interrogation**

14 The trigger options **dchg**, **qchg**, and **dupd** refer to the attribute trigger option (**TrgOps**) of the
15 **DataAttribute** of the common **DATA** classes in IEC 61850-7-3. The trigger options **integrity**
16 and **general-interrogation** shall be trigger conditions defined by the attributes **IntgPd** and **GI**
17 of the **BRCB** respectively.

18 Details related to the generation of a report based on the different trigger options shall be as
19 specified in 14.2.3.2.3.

20 If a **BRCB** does not support one or more of the trigger options, the attempt to set the **TrgOps**
21 attribute to TRUE for one of these not supported values, shall cause a negative response of
22 the **SetReportControlValues** service.

23 A SetReportControlValues of the attribute **TrgOps** shall have the same effect as setting **pur-**
24 **geBuf** to TRUE. if the attribute value changes.

25 **14.2.2.12 IntgPd – integrity period**

26 If **TrgOps** includes a setting indicating **integrity**, the attribute **IntgPd** shall indicate the period
27 in milliseconds used for generating an integrity report. An integrity report shall report the values
28 of **all** members of the related **DATA-SET**. **BufTm** shall have no effect when this change issues
29 a report.

30 If a **BRCB** does not support integrity period then an attempt to set the **IntgPd** attribute to a
31 value greater than 0 shall cause a negative response of the **SetReportControlValues** service.

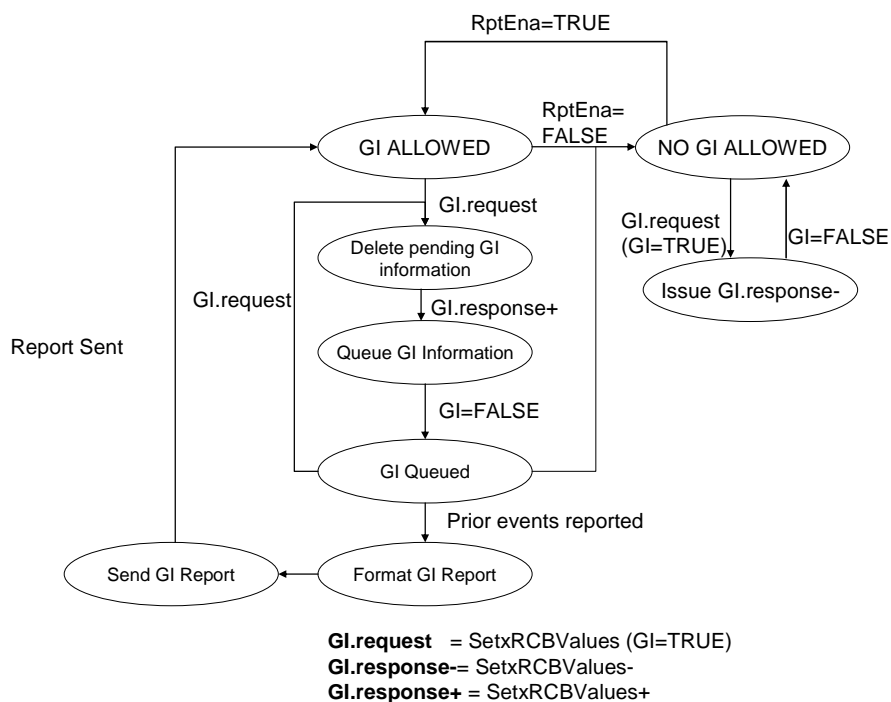
32 A value of 0 shall indicate that no integrity reports shall be issued.

33 A SetReportControlValues of the attribute **IntgPd** shall have the same effect as setting **pur-**
34 **geBuf** to TRUE if the attribute value changes..

35 NOTE An integrity scan may transmit the same values as a general interrogation. The integrity scan is issued pe-
36 riodically by the server. The general-interrogation is issued on demand of the client.

37 **14.2.2.13 GI – general-interrogation**

38 The attribute **GI** shall indicate the request to start the **general-interrogation** process. After
39 setting to TRUE, the **BRCB** shall start the **general-interrogation** process. After initiation of
40 the general interrogation, this attribute shall be automatically set to FALSE by the **BRCB**.



1

2

Figure 25 – Logical State Machine for General Interrogation

3 Figure 25 depicts the Logical State Machine for General Interrogation.

4 The initial state is “NO GI ALLOWED”. Within this state, any GI.request (e.g. GI set TRUE)
 5 shall cause a GI.response- to be sent. Additionally, the implementation shall set the value of GI
 6 to FALSE.

7 A transition of RptEna from FALSE to TRUE causes a state transition from “NO GI ALLOWED”
 8 to “GI ALLOWED”. Within this state, a client may issue a GI.request (e.g. GI transitioning from
 9 FALSE to TRUE). Upon receipt of the GI.request, the implementation for a BRCB shall de-
 10 delete/remove any previously queued general-interrogation information, queue the requested
 11 general-interrogation information, set GI=FALSE, and issue a GI.response+. The order of exe-
 12 cution of these steps is a local issue.

13 NOTE A URCB model does not model a buffer. Therefore, there is no need to delete pending GIs.

14 Normal reporting will eventually cause previously queued events to be reported and the queued
 15 general-interrogation information to be formatted and reported.

16 If a GI.request (e.g. GI transitioning from FALSE to TRUE) is issued while in the GI Queued
 17 state, the general-interrogation process shall be restarted.

18 Once a GI is formatted for reporting, it is a local issue regarding the impact of a new
 19 GI.request. However, once the GI is being sent, the entire set of general-interrogation informa-
 20 tion shall be sent regardless if a new GI.request has been received. In this case, once the
 21 sending of the general-interrogation information is complete, the newly requested general-
 22 interrogation process shall start (e.g. an immediate transition through the GI ALLOWED state).

23 Note While GI=TRUE, subsequent GI.requests shall not result in additional general-interrogation processing.

24 If a client attempts to set GI=FALSE a positive response shall be issued. However, GI pending
 25 processing will not be impacted.

1 If a **BRCB** does not support **general-interrogation** then an attempt to set the **GI** attribute to
2 TRUE shall cause a negative response of the **SetReportControlValues** service containing
3 not-supported.

4 If a BRCB contains TrgOps.GI = FALSE then an attempt to set the GI attribute to TRUE shall
5 cause a SetReportControlValues positive response. No GI report shall be generated.

6 **14.2.2.14 PurgeBuf – purge buffer**

7 The attribute **PurgeBuf** shall indicate the request to discard buffered events. After setting to
8 TRUE, the **BRCB** shall discard all buffered events that have not yet been sent to the client. Af-
9 ter discarding the buffered events, this attribute shall be automatically set to FALSE by
10 the **BRCB**.

11 **14.2.2.15 EntryID – entry identifier**

12 The reported entries are dependant upon the transitions of state of the BRCB:

13 – A transition from disabled to enabled shall start reporting with the first available entry (i.e.
14 oldest) in the queue of entries. Reporting of the next sequential entries shall occur.

15 – A transition from resync to enabled shall start reporting with the next available entry (i.e. in
16 time sequence), in the queue of entries, after the entry associated with the EntryID value
17 set by the client. Reporting of the next sequential entries shall occur.

18 The value of EntryID, returned in a GetBRCBValues response shall be defined as follows:

19 – When the BRCB state is RptEna=FALSE: a GetBRCBValues shall return the EntryID value
20 that represents the last (i.e..newest) entry that has been entered into the buffer.

21 – When the BRCB RptEna=TRUE: The value of EntryID, returned in a GetBRCBValues re-
22 sponse, shall be the EntryID of the last EntryID formatted and queued for transmission.

23 An EntryID value of all zeros(0) is reserved to indicate an empty buffer, no reported EntryID
24 shall have a value of zero(0).

25
26 NOTE 1 Clients needing to resync to the first set of entries in the buffer should use SetBRCBValues with an En-
27 tryID whose value is all zeros(0)

28 NOTE 2 The resync state allows a client to set the **EntryID** to the last value of the **EntryID** received with the
29 last proper report in order to synchronize with the server. Clients that desire to insure that a resync has occurred to
30 the appropriate entry (e.g. in the case of server replacement) should read the BRCB's TimeOfEntry attribute.after
31 setting the EntryID attribute value.

32 **14.2.2.16 TimeOfEntry – time of entry**

33 The attribute **TimeOfEntry** shall be the time at which the internal event notification was re-
34 ceived by the report handler. This value is assigned to a specific EntryID which is also as-
35 signed at the time of internal notification receipt.

36 The value, returned in a GetBRCBValues response, shall provide the time stamp of the EntryID
37 whose value is exposed in the control block. The value exposed for TimeOfEntry, when the
38 value of EntryID is zero(0), is a local issue.

39 **14.2.2.17 ResvTms – reservation time**

40 The values of the attribute ResvTms are defined as follows:

- 1 – A value of -1 shall indicate that the **BRCB** is currently exclusively reserved for a set of specific clients based upon configuration.
- 2
- 3 – A value of zero(0), shall indicated that the **BRCB** is not reserved.
- 4 – A positive value shall indicate that the **BRCB** has been dynamically reserved. The value represents the number of seconds that the reservation will be maintained after association loss. Upon expiration of the reservation, the ResvTms value shall be locally set to a value of zero(0)
- 5
- 6
- 7
- 8 – A **SetBRCBValues** request, for setting **ResvTms**, shall:
- 9 – Generate a negative response if the **BRCB's ResvTms** value is non-zero and if the **SetBRCBValues** request is being issued by another client for whom the BRCB is not reserved.
- 10
- 11
- 12 – Generate a negative response if the **BRCB's ResvTms** value is -1.
- 13 – Generate a negative response if the **ResvTms** value to be set is negative.
- 14 – Generate a positive response if the **BRCB's ResvTms** value is zero(0) and the value being set is zero(0) or positive.
- 15
- 16 – Generate a positive response if the **SetBRCBvalues** request where:
- 17 – the **ResvTms** value, being set, is zero(0) or positive,
- 18 – and is from the client for which the **BRCB** has been reserved via a positive value.

19 14.2.3 BRCB class services

20 14.2.3.1 Overview

21 For **BRCB** the following services are defined:

Service	Description
Report	Send a report
GetBRCBValues	Read an attribute of a BRCB
SetBRCBValues	Write an attribute of a BRCB

22

23 14.2.3.2 Report

24 14.2.3.2.1 Report parameter table

25 The report service shall be used by **BRCB** to send reports from the server to the client.

Parameter name
Request
ReportFormat

26 NOTE The Report service is an unconfirmed service. It consists only of a request service primitive. The **DATA-**
 27 **SET** values are sent from the server to the client. In a SCSM this service may be confirmed at, for example, the
 28 transport layer.

1 **14.2.3.2.2 Request**2 **14.2.3.2.2.1 ReportFormat Syntax**

3 The parameter **ReportFormat** shall specify the information to be included in the report. The
4 structure of the report shall be as specified in Table 24.

5 **Table 24 – Report format specification**

ReportFormat		
Parameter name	Parameter type	Explanation
RptID	VISIBLE STRING65 ^a	Report identification
OptFlds	^a	Optional fields to be included in the report
IF sequence-number = TRUE in optFlds		
SqNum	INT16U	Sequence number
SubSqNum	INT16U	Subsequence number
MoreSegmentsFollow	BOOLEAN	More report segments with the same sequence number follow
IF dat-set-name = TRUE in optFlds		
DatSet	ObjectReference ^a	Data set reference
IF buffer-overflow = TRUE in optFlds		
BufOvfl	BOOLEAN	TRUE shall indicate that a buffer overflow has occurred.
If conf-revision = TRUE in optFlds		
ConfRev	INT32U	
Entry		
IF report-time-stamp = TRUE in optFlds		
TimeOfEntry	EntryTime	
IF entryID = TRUE in optFlds		
EntryID	EntryID	
EntryData [1..n]		
IF data-reference = TRUE in optFlds		
DataRef	ObjectReference	Respective DataAttrRef
Value	(*)	(*) type(s) depend on the definition of common data classes in IEC 61850-7-3
ReasonCode	ReasonForInclusion	If reason-for-inclusion (= TRUE) in optFlds. For the definition see 14.2.3.2.2.10.
^a The type and value of this parameter shall be derived from the respective attribute of the BRCB or URCB .		

6

1 14.2.3.2.2.2 RptID – report ID

2 The parameter **RptID** shall be derived from the respective attribute in the **BRCB**.

3 14.2.3.2.2.3 OptFlds – optional fields to include in report

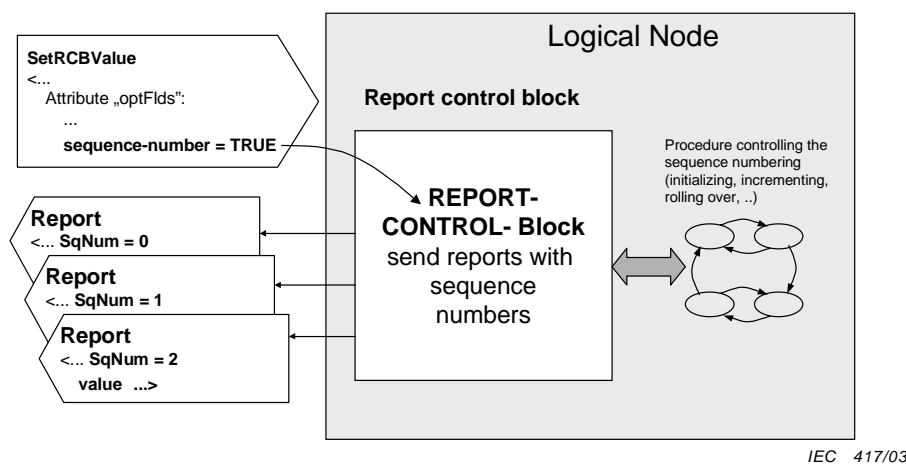
4 The parameter **OptFlds** shall specify which of the optional fields (**sequence-number**, **report-**
5 **time-stamp**, **reason-for-inclusion**, **data-set-name**, **data-reference**, **buffer-overflow**,
6 **entryID**, or **conf-revision**) shall be included in the Report.

7 The parameter **OptFlds** shall be derived from the attribute **OptFlds** of the respective **BRCB**.

8 14.2.3.2.2.4 SqNum – sequence number

9 The **BRCB** that has report enable set to TRUE shall maintain the parameter **SqNum**. This
10 number shall be incremented by the **BRCB** for each report generated and sent on the basis of
11 the **BRCB**. The increment shall occur once the **BRCB** has formatted the report for transmis-
12 sion. The first report following the setting of the report enable to TRUE shall contain sequence
13 number 0. The sequence number shall roll over to 0 at its maximal value.

14 The sequence number shall be included in the report if the optional fields to include in report
15 attribute (**OptFlds**) of the **BRCB** includes the sequence-number (=TRUE); otherwise, it shall be
16 omitted. Figure 26 gives an example of report generation and sequence number.



IEC 417/03

17
18 **Figure 26 – Report example on the use of sequence number**

19 14.2.3.2.2.5 SubSqNum – subsequence number

20 For the case of long reports that do not fit into one message, a single report shall be divided
21 into subreports. Each segment – of one report – shall be numbered with the same sequence
22 number and a unique **SubSqNum**.

23 The **BRCB** shall maintain a subsequence number for each report. This number shall be incre-
24 mented for each subreport generated and sent based upon the report control instance. The in-
25 crement shall occur once the server has formatted the subreports and queued the sub-report to
26 the next lower protocol layer. The first subreport of the report shall have a subsequence num-
27 ber of zero. The subsequence number shall roll over to 0 after all subreports of one specific
28 report have been queued.

29 The subsequence number shall be included in the report if the optional fields to include in re-
30 port attribute (**OptFlds**) of the **BRCB** includes **sequence-number** (=TRUE); otherwise, it shall
31 be omitted.

1 If a **BRCB** does not support sequence numbering then an attempt to set the **sequence-**
 2 **number** of the **OptFlds** attribute to TRUE shall cause a negative response of the
 3 **SetBRCBValues** service.

4 14.2.3.2.2.6 MoreSegmentsFollow – more report segments follow

5 The parameter **MoreSegmentsFollow** indicates that more report segments with the same se-
 6 quence number follow.

7 14.2.3.2.2.7 DatSet – data set reference

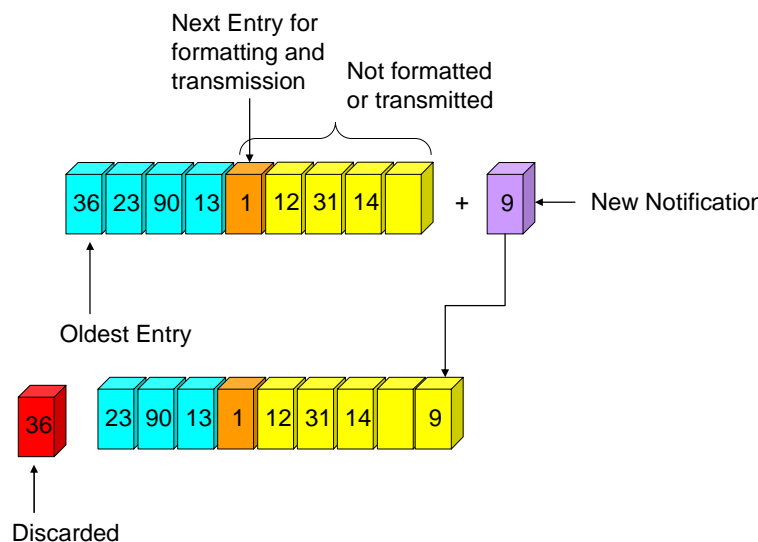
8 The parameter **DatSet** shall be derived from the respective attribute in the **BRCB**.

9 14.2.3.2.2.8 BufOvfl – Possible Information Loss

10 The parameter **BufOvfl** shall indicate to the client that entries within the buffer may have been
 11 lost. The detection of possible loss of information occurs when a client requests a resync to a
 12 non-existent entry or to the first entry in the queue.

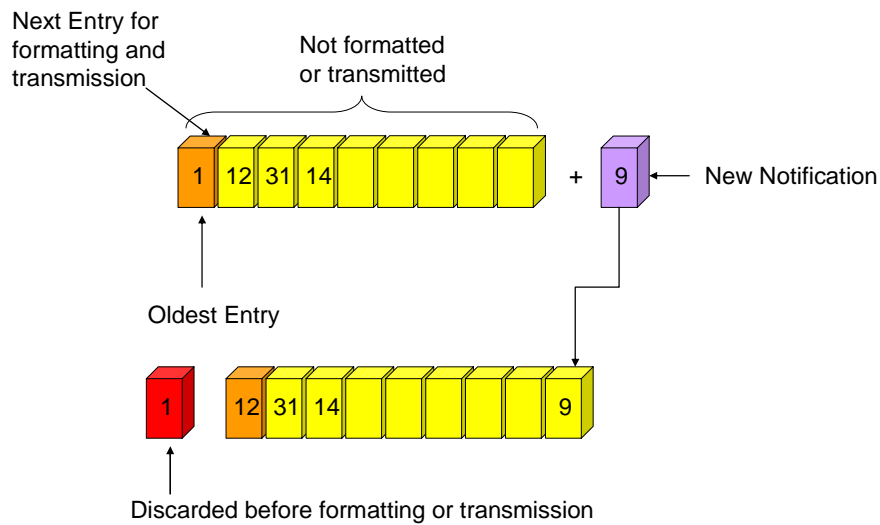
13 The **Report Handler** shall set BufOvfl= TRUE in the first report that is sent after the transition
 14 from disabled to enabled. Subsequent reports shall have BufOvfl = FALSE.

15 Information can also be lost if there are resource constraints that are encountered during the
 16 **enabled** state (e.g. bandwidth, memory due to a high influx of notifications, etc.). Implementa-
 17 tions shall discard the oldest Entry(s) in the queue in order to accept new notifications. If one of
 18 Entries discarded causes the Report Handler to move the pointer to the Next Entry for transi-
 19 smission, the implementation shall indicated BufOvfl=TRUE in the next entry that is formatted
 20 and transmitted only.



21

22 **Figure 27 – Entry discard that does not cause indication of loss of information in en-**
 23 **abled state**



1

2 **Figure 28 – Indication of loss of Information due to resource constraints in Enable State**

3 Figure 27 shows an example where an entry is discarded that has already been formatted and
4 queued for transmission and therefore BufOvfl shall not be set TRUE.

5 Figure 28 shows an example where an entry is discarded that has not been formatted and
6 queued for transmission. Therefore, BufOvfl shall be set TRUE in the next report generated
7 for transmission.

8 **14.2.3.2.2.9 Entry**

9 **TimeOfEntry – report time stamp**

10 The parameter **TimeOfEntry** shall specify the time when the EntryID was created . The
11 **TimeOfEntry** shall be included in the report if the optional fields to include attribute (**OptFlds**)
12 of the **BRCB** includes report-time-stamp (=TRUE), otherwise it shall be omitted.

13 NOTE The event “time at which the report was generated” is determined by a specific implementation.

14 If the **BRCB** does not support **TimeOfEntry** then an attempt to set the report-time-stamp of the
15 **OptFlds** attribute to TRUE shall cause a negative response of the **SetBRCBValues** service.

16 Reports with the same sequence number but different subsequence numbers shall use the
17 same **TimeOfEntry**.

18 **EntryID – entry identifier**

19 For the definition see 14.2.2.15.

20 **EntryData [1..n]**

21 The parameter **EntryData** shall contain the data reference, value, and reasonCode of each
22 member of the **DATA-SET** to be included in the report. The value shall comprise the value of
23 all data attributes of the member of **DATA-SET**.

24 **DataRef**

25 The parameter **DataRef** shall contain the reference of the DataSet member that is being
26 reported.

1 **Value**

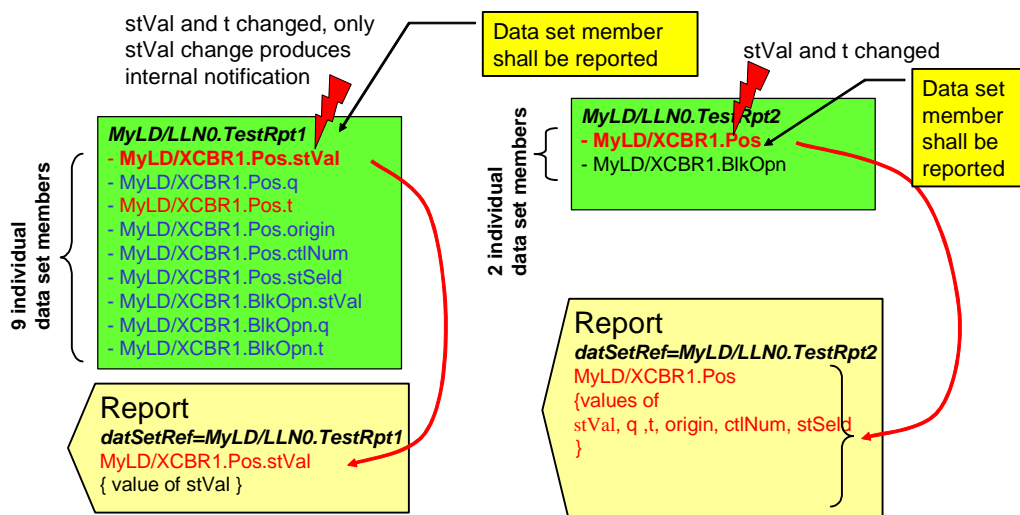
2 The parameter **Value** shall contain the **DataAttribute** values included in the report.

3 The number of members of the **DATA-SET** whose values shall be included in the report
4 shall depend on the control attribute buffer time (**BufTm**) and the occurrences of internal
5 notifications.

6 **BufTm = 0**

7 In case of (**BufTm** = 0) only the value(s) of the member(s) of a **DATA-SET** shall be in-
8 cluded that produced the EntryID.

9 **EXAMPLE** The data attribute **stVal** of the **DATA** **MyLD/XCBR1.Pos** (Position) in Figure 29 is referenced in
10 two different **DATA-SETS**. The figure displays two different instances that reference the data attributes of
11 the position. In the left case the **DATA-SET** references 9 individual **DATA-SET** members (all of functional
12 constraint **ST**): **Pos.stVal** is one of the nine members. In case of the change produced by the **member**
13 **stVal**, the value for exactly that member will be included in the report. The **DATA-SET** in the right example
14 has just two members. The **DATA** **Pos** (which has six data attributes: **stVal**, **q**, **t**, ...) is one of the two mem-
15 bers. A change produced in the **member Pos** (for example, by the change in the **DataAttribute** **stVal**)
16 causes the inclusion of the values of all **DataAttribute** of the **DATA-SET** member **Pos** (i.e., the complete
17 member comprising all six **DataAttributes** **stVal**, **q**, **t**, ...).



18 All data attributes in this example are functionally constrained by FC=ST

IEC 418/03

19 **Figure 29 – Data set members and reporting**

20 **BufTm > 0**

21 In the case of (**BufTm** > 0) the values of all members of a **DATA-SET** containing attrib-
22 utes that have changed with respect to **TrgOps** during the **bufferTime** interval shall be
23 included that produced internal notifications during the **buffer time**. Further constraints
24 apply; see 14.2.2.9 for additional details on **BufTm**.

25 **ReasonCode – reason for inclusion**

26 The reason for inclusion shall be included in the report if the optional fields to include in
27 report attribute (**OptFlds**) of the **BRCB** includes reason for inclusion (=TRUE); otherwise,
28 it shall be omitted. The value for the reason for inclusion shall be set according to the
29 **TrgOps** that caused the creation of the report. The value range for reasons for inclusion
30 shall be as defined in 14.2.3.2.2.10.

31 **14.2.3.2.2.10 ReasonCode (ReasonForInclusion)**

32 The values conveyed by **ReasonForInclusion** shall be a **PACKEDLIST** as defined in Table 25.

1

Table 25 – ReasonForInclusion

ReasonForInclusion			
Attribute Name	Attribute Type	Value/Value Range	M/O/C
	PACKEDLIST		
data-change	BOOLEAN		c1
quality-change	BOOLEAN		c2
data-update	BOOLEAN		c3
integrity	BOOLEAN		c4
general-interrogation	BOOLEAN		c5
c1 – may only be TRUE if TrgOps.dchg = TRUE. c2 – may only be TRUE if TrgOps.qchg = TRUE c3 – may only be TRUE if TrgOps.dupd = TRUE. c4 – may only be TRUE if IntgPd is non-zero and TrgOps.integrity = True. c5 – may only be TRUE if there has been a SetBRCBValues of GI=TRUE and TrgOps.general-interrogation=TRUE.			

2

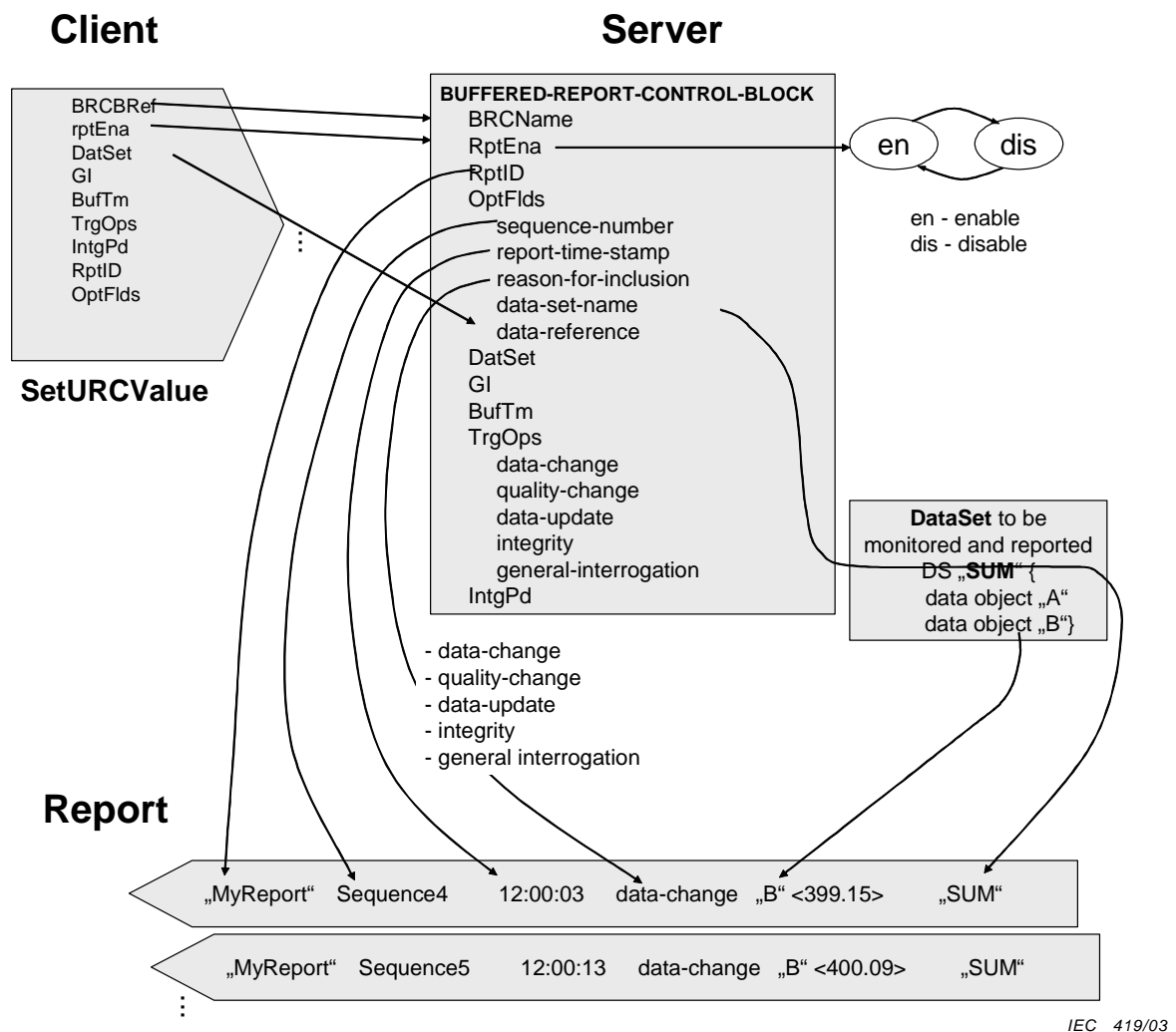
3 General-interrogation and integrity reasons for inclusions are mutually exclusive of all reasons
 4 for inclusion. See 14.2.3.2.3.5.

5 **14.2.3.2.3 Procedures for report generation**

6 **14.2.3.2.3.1 Overview**

7 Figure 30 shows the principle relation between a **BRCB** and the processing of the report.
 8 The information that is to be included in the report and how it is to be included depends on
 9 the attribute settings of the **BRCB**.

10 NOTE Not all attributes and not all details are shown in Figure 30.



IEC 419/03

1
2

Figure 30 – Report example

3 **Pre-condition**

4 A **BRCB** shall have been configured and enabled for reporting and shall have an established
5 association with the client to which the information is to be reported.

6 **14.2.3.2.3.2 Data-change, quality-change, and data-update**

7 These three trigger options support report generation based on change or update in a value of
8 a **DataAttribute** of a member of a **DATA-SET**.

9 **data-change**

10 The trigger option **data-change** (**TrgOps.dchg** is TRUE) relates to a change in a value of a
11 **DataAttribute** representing the process-related value of the data. If the **TrgOps.dchg** is
12 FALSE then no report should be issued on a **data-change** in the value of that **DataAttribute**.

13 **quality-change**

14 The trigger option **quality-change** (**TrgOps.qchg**=TRUE) relates to a change in the quality
15 value of a **DataAttribute**. If the **TrgOps.qchg** is FALSE then no report should be issued on a
16 **quality-change** in the value of that **DataAttribute**.

1 **data-update**

2 The trigger option **data-update** (**TrgOps.dupd=TRUE**) relates to a freeze event in a value of
3 a **DataAttribute** representing a freeze value of the data (for example, frozen counters) or to
4 an event triggered by updating the value of a **DataAttribute**. If the **TrgOps.dupd** is FALSE
5 then no report should be issued on a **data-update** in the value of that **DataAttribute**.

6 NOTE 1 Data-update trigger condition may be used to issue sending a report or storing a log entry into a log when
7 a value of a **DataAttribute** has updated. Updating may mean that the value has changed or has been "overwritten"
8 with the same value as before. The dupd trigger condition can be used as a trigger for statistics values that may be
9 calculated and updated on a periodic base. Independently of whether the statistics value has changed or not, the
10 value will be reported or logged.

11 NOTE 2 With the specification of the common data classes in IEC 61850-7-3, the trigger option applying to
12 a specific **DataAttribute** is defined.

13 When the **BRCB** is notified by an internal notification of a **data-change**, **quality-change**, or
14 **data-update** event of a member(s) of the referenced **DATA-SET** whose values are to be re-
15 ported, the **BRCB** shall include the value of the member(s) of the referenced **DATA-SET** that
16 produced the internal notification in the report according to 14.2.3.2.2.9. The value to be re-
17 ported shall be the value that produced the internal notification.

18 NOTE 3 For changes that meet more than one **TrgOp** criteria (for example, **data-change** and **quality-change**),
19 it is preferable to send only a single report in such a case.

20 **14.2.3.2.3.3 Integrity**

21 The trigger option **integrity** supports integrity report generation. In addition, to activate this
22 trigger option (set **TrgOps integrity** to TRUE), a client shall set the integrity period (**IntgPd**)
23 to a value greater than 0. When integrity reports are enabled, the **BRCB** shall be notified
24 each time the value of the time as specified in **IntgPd** has expired. The **BRCB** shall then
25 build a report with the values of **all** members of the referenced **DATA-SET**. If the **TrgOps**
26 (= **integrity**) is FALSE or if **IntgPd** is zero, no integrity report should be issued.

27 All buffered entries shall be sent before integrity reports can be sent.

28 A new internal notification caused by **data-change**, **quality-change**, or **data-update** (while
29 the transmission of the integrity report is still going on) shall use a new sequence number (and
30 subsequence number starting with 0). No other reports shall be transmitted until the entire in-
31 tegrity report has transmitted.

32 A new notification caused by integrity time (while the transmission of the integrity report is still
33 going on) shall be interpreted as a mis-configured **BRCB**. The new notification shall have no ef-
34 fect.

35 A new **general-interrogation** request (while the transmission of the integrity report is still go-
36 ing on) shall be deferred until the ongoing transmission of the integrity report has completed.
37 A new general-interrogation report with a new sequence number (and subsequence number
38 starting with 0) shall be generated and sent.

39 **14.2.3.2.3.4 GI (General interrogation)**

40 The attribute **general interrogation** (**GI**) shall be used to indicate the request of a general in-
41 terrogation. After setting the attribute **GI** to TRUE the **BRCB** shall start the interrogation proc-
42 ess and create a report that includes all **DataAttribute** values of the referenced **DATA-SET**.
43 After initiation of the interrogation process the **BRCB** shall automatically set the value of **GI** to
44 FALSE. If the **TrgOps.general-interrogation** is FALSE then no general-integrity report
45 should be issued.

46 All buffered entries shall be sent before **general-interrogation** reports can be sent.

1 A new request for **general-interrogation** (while the transmission of the **general-**
2 **interrogation** report is still going on) shall stop sending the remaining segments of the **gen-**
3 **eral-interrogation** report that is still going on. A new **general-interrogation** report with a
4 new sequence number (and subsequence number starting with 0) shall be generated and sent.

5 A new notification caused by integrity time (while the transmission of the **general-**
6 **interrogation** report is still going on) shall be deferred until the ongoing transmission of the
7 **general-interrogation** report has completed.

8 NOTE The **general-interrogation** is initiated by the client. The integrity report, which also transmits all values of
9 a data set, is initiated by the **BRCB**.

10 **14.2.3.2.3.5 Time sequence order of reports**

11 The **BRCB** within the implementation resource limits shall send all reports in the time sequence
12 order in which the related entries have been created.

13 Reports generated as result of the trigger options **integrity** or **general-interrogation** provide
14 a snapshot of the values of **all** members of the **DATA-SET**. The transmission of these reports
15 shall start with the next sequence number. If all values of the referenced data set do not fit into
16 one single report, several subreports with incremented subsequence number (starting with
17 subsequence number equal shall be sent until all values have been sent. If – while sending
18 these reports or subreports respectively – **DATA** values caused by **data-change**, **quality-**
19 **change**, or **data-update** need to be sent, this shall be done with a new report sent after the
20 transmission of the **integrity** or **general-interrogation** report.

21 NOTE This allows a client to keep a process data image consistent when a report is received while a general-
22 interrogation is in progress. The client needs to keep track of the sequence numbers. When receiving information
23 for a specific data in a report with sequence number (for example, 22) older than the sequence number (for exam-
24 ple, 23) of a previously received report with the same data, the client may not use this information to update the
25 process data image.

26 **14.2.3.2.3.6 Buffering events**

27 The **BRCB** shall buffer entries based on the trigger options **data-change**, **quality-change**,
28 **data-update**, and **integrity** during all states: disabled; resync; and enabled

29 After the association is available again, after the client has set the **EntryID**, and enabled the
30 **BRCB**, the **BRCB** shall start sending the reports of entries that have been buffered. The **BRCB**
31 shall use the sequence and subsequence numbers so that no gaps occur.

32 NOTE 1 Since the buffer entries based on the trigger option integrity are buffered by the BRCB, and the memory
33 of the IED dedicated for the buffering is limited, it is recommended to use the trigger option integrity in the BRCB
34 with great care, to avoid a BufOvfl, and keep a long historical of the entries.

35 NOTE 2 Server implementations determine the number of entries that can be buffered. During association loss,
36 the number of entries may increase to a point that upon transition to the **enabled** state, reporting of all the entries
37 may take a long time. Client and server implementations should attempt to address this issue through high-priority
38 formatting and queueing for transmission of the entries or limiting the number of entries.

39

40 **14.2.3.3 GetBRCBValues**

41 **14.2.3.3.1 GetBRCBValues parameter table**

42 A client shall use the **GetBRCBValues** service to retrieve attribute values of **BRCB** made visi-
43 ble and thus accessible to the requesting client by the referenced **LOGICAL-NODE**.

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

3

Parameter name
Request
BRCBReference
FunctionalConstraint
Response+
ReportIdentifier
ReportEnable
DataSetReference
ConfigurationRevision
OptionalFields
BufferTime
SequenceNumber
TriggerOptionsEnabled
IntegrityPeriod
GeneralInterrogation
PurgeBuf
EntryIdentifier
TimeOfEntry
ReserveTimeSecond [0..1]
Response–
ServiceError

4

5 14.2.3.3.2 Request

6 14.2.3.3.2.1 BRCBReference

7 The parameter **BRCBReference** shall specify the ObjectReference of the **BRCB**.

8 The service parameter **BRCBReference** shall be **BRCBRef**.

9 14.2.3.3.2.2 FunctionalConstraint

10 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
11 rameter to filter the respective instances of attributes of a **BRCB**.

12 The service parameter **FunctionalConstraint** shall be **BR**.

13 14.2.3.3.3 Response+

14 The parameter **Response+** shall indicate that the service request succeeded. All SCSMs shall
15 be able to return the Response+ parameter values of:

16 14.2.3.3.3.1 ReportIdentifier

17 The parameter **ReportIdentifier** shall contain the value of the corresponding attribute **RptID**
18 of the referenced **BRCB**.

1 **14.2.3.3.3.2 ReportEnable**

2 The parameter **ReportEnable** shall contain the value of the corresponding attribute **RptEna** of
3 the referenced **BRCB**.

4 **14.2.3.3.3.3 DataSetReference**

5 The parameter **DataSetReference** shall contain the value of the corresponding attribute
6 **DatSet** of the referenced **BRCB**.

7 **14.2.3.3.3.4 ConfigurationRevision**

8 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute
9 **ConfRev** of the referenced **BRCB**.

10 **14.2.3.3.3.5 OptionalFields**

11 The parameter **OptionalFields** shall contain the value of the corresponding attribute **OptFlds**
12 of the referenced **BRCB**.

13 **14.2.3.3.3.6 BufferTime**

14 The parameter **BufferTime** shall contain the value of the corresponding attribute **BufTm** of the
15 referenced **BRCB**.

16 **14.2.3.3.3.7 SequenceNumber**

17 The parameter **SequenceNumber** shall contain the value of the corresponding attribute
18 **SqNum** of the referenced **BRCB**.

19 **14.2.3.3.3.8 TriggerOptionsEnabled**

20 The parameter **TriggerOptionsEnabled** shall contain the value of the corresponding attribute
21 **TrgOps** of the referenced **BRCB**.

22 **14.2.3.3.3.9 IntegrityPeriod**

23 The parameter **IntegrityPeriod** shall contain the value of the corresponding attribute **IntgPd**
24 of the referenced **BRCB**.

25 **14.2.3.3.3.10 GeneralInterrogation**

26 The parameter **GeneralInterrogation** shall contain the value of the corresponding attribute
27 **GI** of the referenced **BRCB**.

28 **14.2.3.3.3.11 PurgeBuf**

29 The parameter **PurgeBuf** shall contain the value of the corresponding attribute **PurgeBuf** of
30 the referenced **BRCB**.

31

32 **14.2.3.3.3.12 EntryIdentifier**

33 The parameter **EntryIdentifier** shall contain the value of the corresponding attribute **EntryID**
34 of the referenced **BRCB**.

1 **14.2.3.3.3.13 TimeOfEntry**

2 The parameter **TimeOfEntry** shall contain the value of the corresponding attribute **TimeOfEntry**
3 **of the referenced BRCB.**

4 **14.2.3.3.3.14 ReserveTimeSecond [0..1]**

5 The parameter **ReserveTimeSecond** shall contain the value of the corresponding attribute
6 **ResvTms** of the referenced **BRCB.**

7 **14.2.3.3.4 Response–**

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

10 **14.2.3.4 SetBRCBValues**

11 **14.2.3.4.1 SetBRCBValues parameter table**

12 A client shall use the **SetBRCBValues** service to set attribute values of **BRCB** made visible
13 **and thus accessible to the requesting client by the referenced LOGICAL-NODE.**

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

Parameter name
Request
BRCBReference
FunctionalConstraint
ReportIdentifier [0..1]
ReportEnable [0..1]
DataSetReference [0..1]
OptionalFields [0..1]
BufferTime [0..1]
TriggerOptionsEnabled [0..1]
IntegrityPeriod [0..1]
GeneralInterrogation [0..1]
PurgeBuffer [0..1]
EntryIdentifier [0..1]
ReserveTimeSecond [0..1]
Response+
Response–
ServiceError

16

17 **SetBRCBValues.request** shall be processed atomically by the server. If the request contains
18 **any parameters, whose sets fails, a Response- shall be returned.**

19

1 **14.2.3.4.2 Request**

2 **14.2.3.4.2.1 BRCBReference**

3 The parameter **BRCBReference** shall specify the ObjectReference of the **BRCB**.

4 The service parameter **BRCBReference** shall be **BRCBRef**.

5 **14.2.3.4.2.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
7 rameter to filter the respective attributes of a **BRCB**.

8 The service parameter **FunctionalConstraint** shall be **BR**.

9 **14.2.3.4.2.3 ReportIdentifier [0..1]**

10 The parameter **ReportIdentifier** shall contain the value for the corresponding attribute **RptID**
11 of the referenced **BRCB**.

12 **14.2.3.4.2.4 ReportEnable [0..1]**

13 The parameter **ReportEnable** shall contain the value for the corresponding attribute **RptEna**
14 of the referenced **BRCB**.

15 **14.2.3.4.2.5 DataSetReference [0..1]**

16 The parameter **DataSetReference** shall contain the value for the corresponding attribute
17 **DatSet** of the referenced **BRCB**.

18 **14.2.3.4.2.6 OptionalFields [0..1]**

19 The parameter **OptionalFields** shall contain the value for the corresponding attribute **OptFlds**
20 of the referenced **BRCB**.

21 **14.2.3.4.2.7 BufferTime [0..1]**

22 The parameter **BufferTime** shall contain the value for the corresponding attribute **BufTm** of
23 the referenced **BRCB**.

24 **14.2.3.4.2.8 TriggerOptionsEnabled [0..1]**

25 The parameter **TriggerOptionsEnabled** shall contain the value for the corresponding attribute
26 **TrgOps** of the referenced **BRCB**.

27 **14.2.3.4.2.9 IntegrityPeriod [0..1]**

28 The parameter **IntegrityPeriod** shall contain the value for the corresponding attribute **IntgPd**
29 of the referenced **BRCB**.

30 **14.2.3.4.2.10 GeneralInterrogation [0..1]**

31 The parameter **GeneralInterrogation** shall contain the value for the corresponding attribute
32 **GI** of the referenced **BRCB**.

1 **14.2.3.4.2.11 PurgeBuffer [0..1]**

2 The parameter **PurgeBuffer** shall contain the value for the corresponding attribute **PurgeBuf**
3 of the referenced **BRCB**.

4 **14.2.3.4.2.12 EntryIdentifier[0..1]**

5 The parameter **EntryIdentifier** shall contain the value of the corresponding attribute **EntryID**
6 of the referenced **BRCB**.

7 **14.2.3.4.2.13 ReserveTimeSecond[0..1]**

8 The parameter **ReserveTimeSecond** shall contain the value of the corresponding attribute
9 **ResvTms** of the referenced **BRCB**.

10 **14.2.3.4.3 Response+**

11 The parameter **Response+** shall indicate that the service request succeeded.

12 **14.2.3.4.4 Response–**

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

15

1 **14.2.4 UNBUFFERED-REPORT-CONTROL-BLOCK (URCB) class definition**2 **14.2.4.1 URCB class Syntax**3 The **URCB** class shall have the structure defined in Table 25.4 **Table 26 – URCB class definition**

URCB class					
Attribute name	Attribute type	FC	TrgOp	r/w	Value/value range/explanation
URCBName	ObjectName	-	-		Instance name of an instance of BRCB
URCBRef	ObjectReference	-	-		Path-name of an instance of BRCB
Specific to report handler					
RptID	VISIBLE STRING65	RP	-	rw	c1
RptEna	BOOLEAN	RP	dchg	rw	
Resv	BOOLEAN	RP	-		
DatSet	ObjectReference	RP	dchg	rw	c1
ConfRev	INT32U	RP	dchg	r	
OptFlds	PACKED LIST	RP	dchg	rw	c1
sequence-number	BOOLEAN				
report-time-stamp	BOOLEAN				
reason-for-inclusion	BOOLEAN				
data-set-name	BOOLEAN				
data-reference	BOOLEAN				
buffer-overflow	BOOLEAN				
entryID	BOOLEAN				c2
conf-revision	BOOLEAN				
BufTm	INT32U	RP	dchg	rw	c1
SqNum	INT8U	RP	-	r	
TrgOps	TriggerConditions	RP	dchg	rw	c1
IntgPd	INT32U	RP	dchg	rw	0.. MAX; 0 implies no integrity report.
GI	BOOLEAN	RP	-	rw	
Services					
Report GetURCBValues SetURCBValues					
Notes and Conditions					
Note: An attribute that is marked “r” indicates that the URCB attribute may be obtained (e.g. read) through the use of the GetBRCBValues service. An attribute that is marked “w” indicates that the URCB attribute may be set (e.g. written) through the use of the SetBRCBValues service.					
c1: These attributes may only be set when RptEna = FALSE. A SetURCBValues of these parameters, when RptEna=TRUE, shall fail.					
c2: It is a local issue if a SetURCBValues with OptFlds.EntryID = TRUE should result with a GetURCBValues of OptFlds.EntryID = TRUE. It is suggested that a GetURCBValues of OptFlds.EntryID return a value of FALSE for OptFlds.EntryID .					

5

6 Except **URCBName**, **URCBRef**, **RptEna**, **SqNum**, and **Resv** all other attributes shall be as defined for the **BRCB** in 14.2.2.
7

1 14.2.4.2 URCBName – unbuffered report control name

2 The attribute **URCBName** shall be the name of the **URCB** that unambiguously identifies the
3 **URCB** within **LOGICAL-NODE**.

4 14.2.4.3 URCBRef – unbuffered report control ObjectReference

5 The attribute **URCBRef** shall be the unique path-name of **URCB**.

6 The **ObjectReference URCBRef** shall be:

LDName/LNName.URCBName

7

8 14.2.4.4 RptEna – report enable

9 The attribute **RptEna** (if set to TRUE) shall indicate that the **URCB** is currently enabled to re-
10 port values of the **DATA-SET**. If set to TRUE, the **URCB** shall monitor the referenced value of
11 the **DATA-SET** and generate the reports as specified in the **URCB**. If set to FALSE the **URCB**
12 shall stop issuing reports.

13 While being TRUE (report enabled), no changes of attribute values of the **URCB** other than
14 disabling and activating the trigger options general-integration shall be allowed.

15 If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client over which **URCB** has been
16 enabled is lost, the server shall set the attribute **RptEna** to FALSE.

17 14.2.4.5 Resv – reserve URCB

18 The attribute **Resv** (if set to TRUE) shall indicate that the **URCB** is currently exclusively reserved
19 for the client that has set the value to TRUE. Other clients shall not be allowed to set any at-
20 tribute of that **URCB**.

21 If the attribute **Resv** is not set to TRUE, then setting the attribute **RptEna** to TRUE reserves
22 the instance implicitly.

23 NOTE The attribute **Resv** functions as a semaphore for the configuration, enabling and disabling of the **URCB**.

24 14.2.5 URCB class services

25 14.2.5.1 Overview

26 For the **URCB** the following services are defined.

Service	Description
Report	Send a report
GetURCBValues	Read an attribute of an instance of URCB
SetURCBValues	Write an attribute of an instance of URCB

27

28 14.2.5.2 Report

29 The report service shall be as defined for **BRCB** in 14.2.3.2, exception of:

30 - the parameters of BufOvfl and EntryID of the report format shall not be available.

- 1 - regardless of the value of OptFlds.EntryID, no EntryID shall be included in the Report.
2 - regardless of the value of OptFlds.BufOvfl, no BufOvfl shall be included in the Report.

3 NOTE It is recommended that URCB implementations, that encounter local resource constraints due to the volume
4 of reports (e.g. congestion, memory), issue an association abort to indicate to the client that information has been
5 lost.

6 **14.2.5.3 GetURCBValues**

7 A client shall use the **GetURCBValues** service to retrieve attribute values of an **URCB** made
8 visible and thus accessible to the requesting client by the referenced **LOGICAL-NODE**.

9 The service shall be as defined in 14.2.3.3, except that the parameter **BRCBReference** shall
10 be **URCBReference**, the parameter **PurgeBuffer** shall not be available, and the parameter
11 **functional constraint** shall be **RP**.

12 **14.2.5.4 SetURCBValues**

13 A client shall use the **SetURCBValues** service to set attribute values of **URCB** made visible
14 and thus accessible to the requesting client by the referenced **LOGICAL-NODE**.

15 The service shall be as defined in 14.2.3.4, except that the parameter **BRCBReference** shall
16 be **URCBReference**, the parameter **PurgeBuffer** shall not be available, and the parameter
17 **functional constraint** shall be **RP**.

18

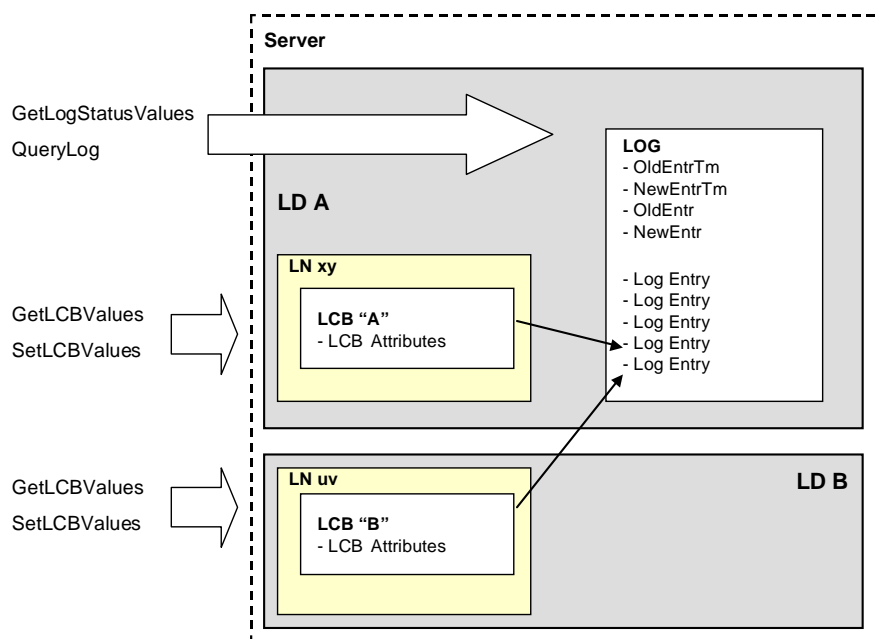
1 14.3 LOG-CONTROL-BLOCK class model

2 14.3.1 General

3 14.3.1.1 Basic concepts

4 Many IEDs have requirements for the internal storage of historical data values and retrieval
 5 over communications systems. This data values fall into two general categories: **periodic re-**
 6 **cordings** (commonly referred to in metering applications as profiles) and **event-triggered or**
 7 **“sequence-of-events” (SOE)** data. Several criteria are be used to differentiate historical data
 8 logging requirements from report-oriented information transfer.

- 9 – Data logging shall be independent of external application associations or other communication
- 10 transactions. Even if communication is lost, historical events shall occur and shall be logged.
- 11 – The process of storing the historical records is completely asynchronous with retrieval over
- 12 communications.
- 13 – The rate of generation of historical records can in some cases be much faster than the abil-
- 14 ity of communication processes to report the values to an external data base.
- 15 – Record retrieval shall allow external applications to request subsets of the entire historical
- 16 data base for the purpose of maintaining an external, complete time or event-sequenced
- 17 historical record.
- 18 – The source of the data may be external to the device. Thus, the historical repository may
- 19 simply be a central point of storage.
- 20 – Records have relative significance with regard to time or ordering and may require the as-
- 21 signment of a sequence number.



IEC 420/03

22
23 **Figure 31 – Log model overview**

24 Figure 31 gives an overview of the **LOG** and **LCB** classes. One **LOG** may be controlled by mul-
 25 tiple **LCBs**.

26 14.3.1.2 The log buffer concept

27 From an implementation view, the **LOG** may be considered as a circular buffer that overwrites
 28 the oldest values in the **LOG**. However, this is hidden from the client. The client view of the
 29 **LOG** is a linear buffer, where the **LOG** entries are identified by

- 1 - **EntryID**: a unique identifier of a **LOG** entry;
- 2 - **TimeOfEntry**: the time, when the **LOG** entry has been added to the **LOG**.

3 **EntryID** shall be a counter that rolls over when the maximal value has been reached. The size
 4 of that counter shall be larger than the maximal number of entries that can be stored in a **LOG**
 5 so that there may not be two entries in the log with the same value of **entryID**. **EntryID** to-
 6 gether with **TimeOfEntry** provide a unique identification of the entry.

7 A client may query the **LOG** by **entryID** or by **TimeOfEntry**.

8 **14.3.2 LCB class definition**

9 **14.3.2.1 LCB class syntax**

10 The **LCB** shall control the procedures that are required for storing values of **DataAttribute**
 11 (the log entry) into a **LOG**. Each enabled **LCB** shall associate **DATA-SET** with a **LOG**. Changes
 12 in a value of a member of a **DATA-SET** shall be stored as LOG entry. Multiple **LCBs** allow mul-
 13 tiple **DATA-SETS** to feed a **LOG**.

14 It shall be the responsibility of access control, to prevent unauthorized clients to modify an **LCB**.

15 NOTE The internal notification, local storage mechanism, internal formats, etc. for log entries are all local issues
 16 and outside the scope of this part of IEC 61850.

17 The **LCB** shall have the structure specified in Table 27.

18 **Table 27 – LCB class definition**

LCB class				
Attribute name	Attribute type	FC	TrgOp	Condition
LCBName	ObjectName	-	-	Instance of LCB
LCBRef	ObjectReference	-	-	Path-name of an instance of LCB
Specific to log handler				
LogEna	BOOLEAN	LG	dchg	
DatSet	ObjectReference	LG	dchg	
OptFlds	PACKED LIST	LG	dchg	
reason-for-inclusion	BOOLEAN			
TrgOps	TriggerConditions	LG	dchg	Valid values for TrgOps of type TriggerConditions shall be dchg, qchg, dupd, and integrity.
IntgPd	INT32U	LG	dchg	1..MAX; 0 implies no integrity logging.
Specific to building the log				
LogRef	ObjectReference	LG		
Services				
GetLCBValues				
SetLCBValues				

Why do we NOT need the CofRev for Logging? Eventuell nicht notwendig, weil man ja neuerdings mit IncludeAll die DataSeT Reference mit abspeichern kann.

IncludeAll

BufTm

19
 20 **14.3.2.2 LCB class attributes**

21 **14.3.2.2.1 LCBName – log control name**

22 The attribute **LCBName** shall unambiguously identify a **LCB** within the scope of a **LOGICAL-**
 23 **NODE**.

1 **14.3.2.2.2 LCBRef – log control ObjectReference**

2 The attribute **LCBRef** shall be the unique path-name of a **LCB**.

3 The ObjectReference **LCBRef** shall be:

LDName/LNName.LCBName

4

5 **14.3.2.2.3 LogEna – log enable**

6 The attribute **LogEna** shall indicate that this **LCB** is recording into the **LOG** specified by
7 **LogRef**.

8 A transition of **LogEna** from disabled to enabled or from enabled to disabled shall cause a log
9 entry to be placed into the **LOG**.

10 NOTE The attribute LogEna may be set to TRUE automatically by a server after turning the server on.

11 While in the state enabled no changes of attribute values of **LCB** other than disabling shall
12 be allowed.

13 **14.3.2.2.4 DatSet – data set reference**

14 The attribute **DatSet** shall indicate the **DATA-SET**, whose member values are to be logged.

15 **14.3.2.3 OptFlds – optional fields to include in log**

16 The attribute **OptFlds** shall be the client-specified optional fields to be included in the log is-
17 sued by this **LCB**. This attribute defines a subset of the optional header fields of the log **Entry-**
18 **Data** (see 14.3.3.1) that shall be included in the log:

19 – reason-for-inclusion (if TRUE **ReasonCode** shall be included in the log);

20 If a **LCB** does not support the above option, then an attempt to set the corresponding bit to
21 TRUE shall cause a negative response of the **SetLCBValues** service.

22 **14.3.2.3.1 TrgOps – trigger options**

23 The attribute **TrgOps** shall specify the trigger conditions that shall be monitored by this **LCB** to
24 cause a Log entry to be created. The values defined are the same as for reporting (see
25 14.2.2.11).

26 The **TrgOps general-interrogation** shall not be supported for logging.

27 **14.3.2.3.2 IntgPd – integrity period**

28 If **TrgOps** is set to integrity, the attribute **IntgPd** indicates the period in milliseconds used for
29 logging caused by integrity scans.

30 **14.3.2.3.3 LogRef – log reference**

31 The attribute **LogRef** shall be the reference of the **LOG** to which values of members of the ref-
32 erenced **DATA-SET** shall be recorded.

1 14.3.2.4 LCB services – Overview

2 For the **LCB** the following services are defined:

Service	Description
GetLCBValues	Retrieve the attribute values of a LCB
SetLCBValues	Set the attributes values of a LCB

3

4 14.3.2.5 GetLCBValues

5 A client shall use the **GetLCBValues** service to retrieve attribute values of **LCB** made visible
6 and thus accessible to the requesting client by the referenced **LOGICAL-NODE**.

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

Parameter name
Request
LCBReference
FunctionalConstraint
Response+
LogEnable
DataSetReference
TriggerOptions
IntegrityPeriod
LogReference
Response–
ServiceError

9

10 14.3.2.5.1 Request

11 14.3.2.5.1.1 LCBReference

12 The parameter **LCBReference** shall specify the **ObjectReference** of the **LCB**.

13 The service parameter **LCBReference** shall be **LCBRef**.

14 14.3.2.5.1.2 FunctionalConstraint

15 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
16 rameter to filter the respective instances of attributes of a **LCB**.

17 The service parameter **FunctionalConstraint** shall be **LG** (logging).

18 14.3.2.5.2 Response+

19 The parameter **Response+** shall indicate that the service request succeeded.

20 14.3.2.5.2.1 LogEnable

21 The parameter **LogEnable** shall contain the value for the corresponding attribute **LogEna** of
22 the referenced **LCB**.

1 **14.3.2.5.2.2 DataSetReference**

2 The parameter **DataSetReference** shall contain the value for the corresponding attribute
3 **DatSet** of the referenced **LCB**.

4 **14.3.2.5.2.3 TriggerOptions**

5 The parameter **TriggerOptions** shall contain the value for the corresponding attribute **TrgOps**
6 of the referenced **LCB**.

7 **14.3.2.5.2.4 IntegrityPeriod**

8 The parameter **IntegrityPeriod** shall contain the value for the corresponding attribute **IntgPd**
9 of the referenced **LCB**.

10 **14.3.2.5.2.5 LogReference**

11 The parameter **LogReference** shall contain the value for the corresponding attribute **LogRef**
12 of the referenced **LCB**.

13 **14.3.2.5.3 Response–**

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

16 **14.3.2.6 SetLCBValues**

17 A client shall use the **SetLCBValues** service to set attribute values of **LCB** made visible and
18 thus accessible to the requesting client by the referenced **LOGICAL-NODE**

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

Parameter name
Request
LCBReference
FunctionalConstraint
LogEnable [0..1]
DataSetReference [0..1]
OptionalFields [0..1]
IntegrityPeriod [0..1]
LogReference [0..1]
Response+
Response–
ServiceError

21

22 SetLCBValues.request shall be processed atomically by the server. If the request contains any
23 parameters, whose sets fails, a Response- shall be returned.

24

1 **14.3.2.6.1 Request**

2 **14.3.2.6.1.1 LCBReference**

3 The parameter **LCBReference** shall specify the ObjectReference of **LCB**.

4 The service parameter **LCBReference** shall be **LCBRef**.

5 **14.3.2.6.1.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
7 rameter to filter the respective instances of attributes of a **LCB**.

8 The service parameter **FunctionalConstraint** shall be **LG** (logging).

9 **14.3.2.6.1.3 LogEnable [0..1]**

10 The parameter **LogEnable** shall contain the value of the corresponding attribute **LogEna** of the
11 referenced **LCB**.

12 **14.3.2.6.1.4 DataSetReference [0..1]**

13 The parameter **DataSetReference** shall contain the value of the corresponding attribute
14 **DatSet** of the referenced **LCB**.

15 **14.3.2.6.1.5 OptionalFields [0..1]**

16 The parameter **OptionalFields** shall contain the value of the corresponding attribute **OptFlds**
17 of the referenced **LCB**.

18 **14.3.2.6.1.6 IntegrityPeriod [0..1]**

19 The parameter **IntegrityPeriod** shall contain the value of the corresponding attribute **IntgPd**
20 of the referenced **LCB**.

21 **14.3.2.6.1.7 LogReference [0..1]**

22 The parameter **LogReference** shall contain the value of the corresponding attribute **LogRef** of
23 the referenced **LCB**.

24 **14.3.2.6.2 Response+**

25 The parameter **Response+** shall indicate that the service request succeeded.

26 **14.3.2.6.3 Response–**

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

29 This service shall return a failure if the service has been issued for any attribute of a **LCB** other
30 than enable while **LCB** is enabled.

31 **14.3.3 LOG class definition**

32 **14.3.3.1 LOG class syntax**

33 The **LOG** shall be filled on a first-in first-out basis. When the list of log entries reaches a point
34 where the stored data reaches the maximal size of the log, the oldest log entry shall be over-

1 written. This action shall have no impact to the further incrementing of the **EntryID** of the
 2 added log entries.

3 The **LOG** shall have the structure defined in Table 28.

4 **Table 28 – LOG class definition**

LOG class			
Attribute name	Attribute type	FC	Value/value range/explanation
LogName	ObjectName		Instance name of an instance of LOG
LogRef	ObjectReference		Path-name of an instance of LOG
OldEntrTm	TimeStamp	LG	
NewEntrTm	TimeStamp	LG	
OldEntr	INT32U	LG	
NewEntr	INT32U	LG	
Entry [1..n]			
TimeOfEntry	EntryTime		
EntryID	EntryID		
EntryData [1..n]			
DataRef	ObjectReference		
Value	(*)		(*) type(s) depend on the definition of common data classes in IEC 61850-7-3
ReasonCode	ReasonForInclusion		If reason-for-inclusion (=TRUE) in optFlds. ReasonCode general-interrogation shall never occur as TRUE. For the definition see 14.2.3.2.2.10.
Services			
QueryLogByTime			
QueryLogAfter			
GetLogStatusValues			

Does not make sense. Could not be used!

5

6 **14.3.3.2 LOG class attributes**

7 **14.3.3.2.1 LogName – log name**

8 The attribute **LogName** shall unambiguously identify a **Log** within the scope of a **Logical**
 9 **Node**.

10 **14.3.3.2.2 LogRef – log reference**

11 The attribute **LogRef** shall be the unique path-name of a **LOG**.

12 The **ObjectReference LogRef** shall be:

LDName/LNName.LogName

13

1 **14.3.3.2.3 OldEntrTm – oldest log entry time of log**

2 The attribute **OldEntrTm** shall indicate the time when the oldest log entry has been stored.

3 NOTE That is the time when the entry has been stored in the log. This is different from the time stamp of the entry
4 itself, which indicates when the event that caused the creation of the log entry has occurred.

5 **14.3.3.2.4 NewEntrTm – newest log entry time of log**

6 The attribute **NewEntrTm** shall indicate the time when the newest log entry has been stored.

7 **14.3.3.2.5 OldEntr – oldest log entry sequence number**

8 The attribute **OldEntr** shall indicate the **EntryID** for the oldest entry available in the log.

9 **14.3.3.2.6 NewEntr – newest log entry sequence number**

10 The attribute **NewEntr** shall indicate the **EntryID** for the newest entry available in the log.

11 **14.3.3.2.7 Entry [1..n]**

12 **14.3.3.2.7.1 TimeOfEntry – time of log entry**

13 The attribute **TimeOfEntry** shall be the time, when the log entry is added to a **LOG**. That time
14 may be different to the time stamp of the data, which shall be the time when the event occurred
15 that caused the log entry to be created.

16 **14.3.3.2.7.2 EntryID – entry identifier**

17 The attribute **EntryID** shall be a unique reference to all log entries having the same value of
18 TimeOfEntry.

19 **14.3.3.2.7.3 EntryData [1..n] – Data of Entry**

20 The parameter **EntryData** shall contain the data reference, values, and reasonCode of each
21 member of the **DATA-SET** to be included in the log entry. The value shall comprise the values
22 of all data attributes of the member of **DATA-SET**.

23 **DataRef**

24 The parameter **DataRef** shall contain contain the DataSet member reference of the
25 value of the EntryData.

26 **Value**

27 The parameter value shall contain the DataSet member values to be included in the En-
28 tryData. There shall be one and only one EntryData per DataRef.

29 The number of members of the **DATA-SET** whose values shall be included in the log
30 shall depend on the **TrgOps** of the **LCB** selected and the following values of **TrgOps** of
31 the respective **DataAttributes**:

32 In case of **TrgOps** (**dchg**, **qchg**, and **data-update**) only the value of the member of a
33 **DATA-SET** shall be included in the log entry that produced the internal event.

34 In case of setting the **LCB** attribute **IntPd** to a value greater than zero (0) and **TrgOp in-**
35 **tegrity** (=TRUE) all values of all members of a **DATA-SET** shall be included in the log
36 entry that produced the internal event.

1 ReasonCode – reason for inclusion

2 The reason for inclusion shall be set according the **TrgOps** that caused the creation of
3 the **EntryData**. The value for reason for inclusion shall be set according the **TrgOps** that
4 caused the creation of the report. The value range for reasons for inclusion shall be
5 as listed:

- 6 – **data-change** (caused by TrgOps = **dchg** in an instance of **DATA**)
- 7 – **quality-change** (caused by TrgOps = **qchg** in an instance of **DATA**)
- 8 – **data-update** (caused by TrgOps = **dupd** in an instance of **DATA**)
- 9 – **integrity** (caused by the attribute **IntgPd** in the **LCB**)

10 14.3.4 Procedures to generate the log entries

11 14.3.4.1 Overview

12 Basically, the conditions and constraints for log generation shall be the same as for report gen-
13 eration (see 14.2.3.2.3). Subclause 14.3.4 specifies the differences only.

14 14.3.4.2 Trigger Options data-change, quality-change, or data-update

15 When the **LCB** is notified by an internal event of a data-change, a quality-change, a data-
16 update of the referenced member of a **DATA-SET**, the **LCB** shall create a LOG entry with the
17 value of the member of **DATA-SET** that produced the internal event.

18 14.3.4.3 Trigger options integrity

19 When a **LCB** is notified as a result of the trigger options integrity, the **LCB** shall create a LOG
20 entry for each member of the referenced **DATA-SET**.

21 14.3.5 LOG services

22 14.3.5.1 Overview

23 For the **LOG** model, the following services are defined:

Service	Description
QueryLogByTime	Read the log entries selected by time
QueryLogAfter	Read the log entries selected by entryID
GetLogStatusValues	Get the status values of a LOG

24

25 14.3.5.2 QueryLogByTime

26 14.3.5.2.1 QueryLogByTime parameter table

27 A client shall use the **QueryLogByTime** service to retrieve a range of **LOG** entries from a **LOG**
28 based on time ranges (**RangeStartTime** and **RangeStopTime**).

Parameter name
Request
LogReference
RangeStartTime
RangeStopTime
Response+
ListOfLogEntries
Response-
ServiceError

29

1 **14.3.5.2.2 Request**2 **14.3.5.2.2.1 LogReference**

3 The parameter **LogReference** shall contain the **ObjectReference LogRef** of a **LOG**. The **Ob-**
 4 **jectReference LogReference** shall be:

LDName/LNName.LogName

5 **14.3.5.2.2.2 RangeStartTime**

6 The parameter **RangeStartTime** shall contain the range start time to retrieve log entries. The
 7 first log entry selected shall be the first entry in the log with a **RangeStartTime** greater than,
 8 or equal to, the **RangeStartTime**. In the case where no **RangeStartTime** is specified,
 9 the first log entry contained in the log shall be the first entry selected for transmission.

10 **14.3.5.2.2.3 RangeStopTime**

11 The parameter **RangeStopTime** shall contain the range stop time to retrieve log entries. The
 12 last log entry selected shall be the last entry in the log with a **RangeStopTime** less than, or
 13 equal to, the **RangeStopTime**. For the case where no **RangeStopTime** is specified, the last
 14 log entry contained in the log shall be the last entry selected.

15 **14.3.5.2.3 Response+**16 **ListOfLogEntries**

17 The parameter **ListOfLogEntries** shall contain the list of log entries that are in the range as
 18 specified with the parameters **RangeStartTime** and **RangeStopTime** of the service request.

19 **14.3.5.2.4 Response–**

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

22 **14.3.5.3 QueryLogAfter**23 **14.3.5.3.1 QueryLogAfter parameter table**

24 A client shall use the **QueryLogAfter** service to retrieve a range of **LOG** entries from the ref-
 25 erenced **LOG** based on ranges of IDs that are after the **RangeStartTime** and **Entry**.

Parameter Name
Request
LogReference
RangeStartTime
Entry
Response+
ListOfLogEntries
Response–
ServiceError

26

1 **14.3.5.3.2 Request**2 **14.3.5.3.2.1 LogReference**

3 The parameter **LogReference** shall specify the **ObjectReference LogRef** of the **LOG**. The
4 **ObjectReference LogReference** shall be:

LDName/LNName.LogName

5

6 **14.3.5.3.2.2 RangeStartTime**

7 The parameter **RangeStartTime** shall contain the time of the log entry (or log entries – in
8 case of multiple entries for a single time stamp) selected.

9 **14.3.5.3.2.3 Entry**

10 The parameter **Entry** shall reference the **LOG** entry of the selected **RangeStartTime** after
11 which the log entries shall be selected.

12 **14.3.5.3.3 Response+**13 **14.3.5.3.3.1 ListOfLogEntries**

14 The parameter **ListOfLogEntries** shall contain the list of log entries that follow after the en-
15 tries as specified with the parameters **RangeStartTime** and **Entry** of the service request.

16 **14.3.5.3.4 Response–**

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

19 **14.3.5.4 GetLogStatusValues**

20 A client shall use the **GetLCBValues** service to retrieve the attribute values of a **LOG** made
21 visible and thus accessible to the requesting client by the referenced **LLNO**.

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

Parameter name
Request
LogReference
FunctionalConstraint
Response+
OldestEntryTime
NewestEntryTime
OldestEntry
NewestEntry
Response–
ServiceError

24

1 **14.3.5.4.1 Request**2 **14.3.5.4.1.1 LogReference**

3 The parameter **LogReference** shall specify the **ObjectReference** of the **LOG**.

4 The service parameter **LogReference** shall be:

LDName/LNName.LogName

Needs to be changed to Functional Constraint Controlblock (FCCB) ... here and on all occurrences of FCCBs (... Report, Log, LCB,

5 **14.3.5.4.1.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint** shall contain the value of the functional constraint parameter to filter the respective instances of attributes of a **LOG**.

8 The service parameter **FunctionalConstraint** shall be **LG** (logging).

9 **14.3.5.4.2 Response+**

10 The parameter **Response+** shall indicate that the service request succeeded.

11 **14.3.5.4.2.1 OldestEntryTime**

12 The parameter **OldestEntryTime** shall contain the value for the corresponding attribute **OldEntrTm** of the referenced **LOG**.

14 **14.3.5.4.2.2 NewestEntryTime**

15 The parameter **NewestEntryTime** shall contain the value for the corresponding attribute **NewEntrTm** of the referenced **LOG**.

17 **14.3.5.4.2.3 OldestEntry**

18 The parameter **OldestEntry** shall contain the value for the corresponding attribute **OldEntr** of the referenced **LOG**.

20 **14.3.5.4.2.4 NewestEntry**

21 The parameter **NewestEntry** shall contain the value for the corresponding attribute **NewEntr** of the referenced **LOG**.

23 **14.3.5.4.3 Response–**

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

26

1 **15 Generic substation event class model (GSE)**

2 **15.1 Overview**

3 The generic substation event model provides the possibility for a fast and reliable system-wide
 4 distribution of input and output data values. The generic substation event model is based on
 5 the concept of an autonomous decentralization, providing an efficient method allowing the si-
 6 multaneous delivery of the same generic substation event information to more than one phys-
 7 cal device through the use of multicast/broadcast services.

8 For the purposes of the generic substation event model, conveyed values are seen from the
 9 viewpoint of the reporting logical device.

10 NOTE 1 It is a matter for the mapping and implementation how reliability and a short transmission delay are
 11 achieved. Depending on the SCSM and communication stack being used, different methods may be implemented.

12 The generic substation event model applies to the exchange of values of a collection of
 13 **DataAttribute**. Two control classes and the structure of two messages are defined in
 14 this clause:

- 15 – generic object oriented substation event (**GOOSE**) supports the exchange of a wide range
 16 of possible common data organized by a **DATA-SET**.
- 17 – generic substation state event (**GSSE**) provides the capability to convey state change in-
 18 formation (bit pairs).

19 NOTE 2 The **GSSE** represents the **GOOSE** model as defined in UCA™ Version 2.

20 The information exchange is based on a publisher/subscriber mechanism. The publisher writes
 21 the values in a local buffer at the sending side; the receiver reads the values from a local buffer
 22 at the receiving side. The communication system is responsible to update the local buffers of
 23 the subscribers. A generic substation event control class in the publisher is used to control the
 24 procedure.

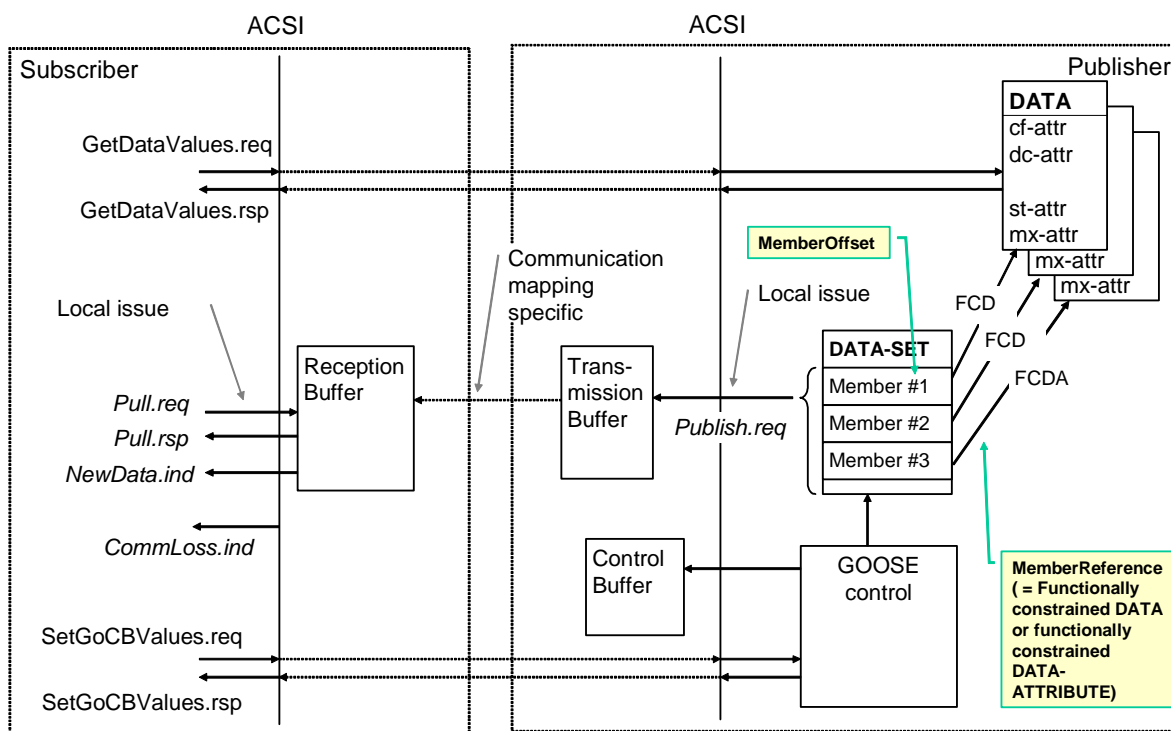


Figure 32 – GoCB model

1 Figure 32 gives an overview of the classes and services of the **GOOSE** model. The message
 2 exchange is based on the multicast application association. If the value of one or several
 3 **DataAttributes** of a specific functional constraint (for example, st) in the **DATA-SET** changes,
 4 the transmission buffer of the publisher is updated with the local service “publish” and the val-
 5 ues are transmitted with a **GOOSE** message. The **DATA-SET** may have several members
 6 (numbered from 1 up – the numbers shall be called **MemberOffset**). Each member shall have
 7 a **MemberReference** referencing the **DataAttribute** with a specific functional constraint (**FC**).
 8 Mapping specific services of the communication network will update the content of the buffer in
 9 the subscribers. New values received in the reception buffer are signalled to the application.

10 The **GOOSE** messages contain information that allow the receiving device to know that a status
 11 has changed and the time of the last status change. The time of the last status change allows a
 12 receiving device to set local timers relating to a given event.

13 A newly activated device, upon power-up or reinstatement to service, shall send current data
 14 (status) or values as the initial **GOOSE** message. Moreover, all devices sending **GOOSE** mes-
 15 sages shall continue to send the message with a long cycle time, even if no status/value
 16 change has occurred. This ensures that devices that have been activated recently will know the
 17 current status values of their peer devices.

18 NOTE 3 The **GSSE** model is similar to the **GOOSE** model. The basic concept described above applies also to the
 19 GSSE model. One major difference is the kind of information exchanged. **GOOSE** provides a flexible means to
 20 specify which information is to be exchanged (**DATA-SET**) whereas **GSSE** provides a simple list of status informa-
 21 tion.

22 The behaviour of the **GoCB** shall apply to the **GsCB**.

23 15.2 GOOSE-CONTROL-BLOCK (GoCB) class

24 15.2.1 GoCB definition

25 The **GoCB** shall be as defined in Table 29.

26 **Table 29 – GOOSE control block class definition**

GoCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
GoCBName	ObjectName	GO	-	Instance name of an instance of GoCB
GoCBRef	ObjectReference	GO	-	Path-name of an instance of GoCB
GoEna	BOOLEAN	GO	dchg	Enabled (TRUE) disabled (FALSE)
GoID	VISIBLE STRING65	GO		Attribute that allows a user to assign a identifica- tion for the GOOSE message. DEFAULT GoCBRef
DatSet	ObjectReference			
ConfRev	INT32U	GO	dchg	
NdsCom	BOOLEAN	GO	dchg	
DstAddress	PHYCOMADDR	GO	dchg	
Services				
SendGOOSEMessage				
GetGoReference				
GetGOOSEElementNumber				
GetGoCBValues				
SetGoCBValues				

Value is split: See new table in 7-3.

27

28 15.2.1.1 GoCBName – GOOSE control block name

29 The attribute **GoCBName** shall unambiguously identify a **GoCB** within the scope of a **LLNO**.

1 15.2.1.2 GoCBRef – GOOSE control block reference

2 The attribute **GoCBRef** shall be the unique path-name of a **GoCB** within the **LLNO**.

3 The **ObjectReference GoCBRef** shall be:

LDName/LLNO.GoCBName

4 15.2.1.3 GoEna – GOOSE enable

5 The attribute **GoEna** (if set to TRUE) shall indicate that the **GoCB** is currently enabled to send
6 GOOSE messages. If set to FALSE the **GoCB** shall stop sending **GOOSE** messages.

7 If there are inconsistent attribute values in the GoCB (e.g. the value of DatSet is Null) or if the
8 value of ConfRev equals 0, a SetGoCBValues with the parameter GoEna equals TRUE shall
9 fail and a negative response shall be issued.

10 While being TRUE (**GoCB** enabled), no changes of attribute values of the **GoCB** other than
11 disabling shall be allowed.

12 15.2.1.4 GoID – GOOSE identifier

13 The attribute **GoID** shall be a user definable identification of the GOOSE message. The default
14 value shall be that of the ObjectReference **GoCB**. However, the value may be set to another
15 value as part of a system wide configuration

16 NOTE Depending upon the SCSM and actual implementation, it may not be possible to uniquely identify the
17 **GOOSE** control or **GSSE** control through the control block reference. Therefore, a standardized control attribute
18 must be provided to allow the system configuration process to be able to uniquely identify the control within the
19 scope of the substation.

20 15.2.1.5 DatSet – data set reference

21 The attribute **DatSet** shall represent the reference of the **DATA-SET** whose values of members
22 shall be transmitted. The members of the **DATA-SET** shall be uniquely numbered beginning
23 with 1. This number is called the **MemberOffset** of a given member. Each member of the
24 **DATA-SET** has a unique number and a **MemberReference** (the functional constraint **DATA**
25 **(FCD)** or **DataAttribute (FCDA)**).

26 NOTE The service **GetGoReference** retrieves the **FCD/FCDA** for a given number, and the service **Get-**
27 **GOOSEElementNumber** retrieves a number for a given **FCD/FCDA**.

28 The initial value of the referenced members of the **DATA-SET** shall be a local issue.

29 15.2.1.6 ConfRev – configuration revision

30 The attribute **ConfRev** shall represent a count of the number of times that the configuration of
31 the **DATA-SET** referenced by **DatSet** has been changed. Changes that shall be counted are:

- 32 – any deletion of a member of the **DATA-SET**;
- 33 – the reordering of members of the **DATA-SET**; and
- 34 – changing the value of the attribute **DatSet**.

35 The counter shall be incremented when the configuration changes. At configuration time, the
36 configuration tool will be responsible for incrementing/maintaining the ConfRev value. When
37 configuration changes occur due to SetGoCBValues, the IED shall be responsible for incre-
38 menting the value of ConfRev.

1 If the value of **DatSet** is set through a **SetGoCBValue** service to the same value, the **ConfRev**
2 value shall still be incremented.

3 The initial value for **ConfRev** is outside the scope of this part of IEC 61850. The value of 0
4 shall be reserved. The value of **ConfRev**, upon a restart of the IED, is a local issue.

5 **15.2.1.7 NdsCom – needs commissioning**

6 The attribute **NdsCom** shall have a value of TRUE if the **GoCB** requires further configuration.

7 Examples, where further configuration is required are:

8 - the attribute **DataSet** has a value of NULL

9 - The number or size of values being conveyed by the elements in the **DatSet** refer-
10 enced **DATA-SET** exceeds constraint determined by the SCSM or the implementation,

11 **15.2.1.8 DstAddress**

12 The attribute **DstAddress** shall be the SCSM specific addressing information like media access
13 address, priority, and other information.

14 **15.2.2 GOOSE service Definitions**

15 **15.2.2.1 Overview**

16 For the **GoCB** the following services are defined:

Service	Description
SendGOOSEMessage	Send GOOSE message
GetGoReference	Retrieve the FCD/FCDA and DatSetReference of a specific member of DATA-SET associated with the GOOSE message
GetGOOSEElementNumber	Retrieve the position of the member in the DATA-SET associated with the GOOSE message of a FCD/FCDA
GetGoCBValues	Retrieve the attributes of a GoCB
SetGoCBValues	Write the attributes of a GoCB

17

18 NOTE - The services **GetGoReference** and **GetGOOSEElementNumber** are used to validate the actual configuration
19 of the publisher versus what the subscriber is expecting to receive. These services provide an alternate solution
20 than reading **GoCB** and **DATA-SET** definitions and can be used by SCSMs for alternate mappings.

21 **15.2.2.2 SendGOOSEMessage**

22 **15.2.2.2.1 SendGOOSEMessage parameter table**

23 The **SendGOOSEMessage** service shall be used by a **GoCB** to send a **GOOSE** message over
24 a **MULTICAST-APPLICATION-ASSOCIATION**.

Parameter name
Request
GOOSE message

25

1 **15.2.2.2.2 Request**2 **GOOSE message**

3 The parameter **GOOSE message** shall specify the **GOOSE** message as defined in 15.2.3 of
4 the given **GoCB**.

5 **15.2.2.3 GetGoReference**6 **15.2.2.3.1 GetGoReference parameter table**

7 A client shall use the **GetGoReference** service to retrieve the **MemberReferences** of specific
8 members of the **DATA-SET** of the referenced **GoCB**.

Parameter name
Request
GoCBReference
MemberOffset [1..n]
Response+
GoCBReference
ConfigurationRevision
DatSet
MemberReference [1..n]
Response–
ServiceError

9

10 **15.2.2.3.2 Request**11 **15.2.2.3.2.1 GoCBReference**

12 The parameter **GoCBReference** shall identify the attribute **GoCBRef** of the **GoCB** for which
13 **MemberReferences** are being requested.

14 **15.2.2.3.2.2 MemberOffset [1..n]**

15 The parameter **MemberOffset** shall contain a number identifying a member of the **DATA-SET**
16 referenced by the attribute **DatSet**.

17 **15.2.2.3.3 Response+**18 **15.2.2.3.3.1 GoCBReference**

19 The parameter **GoCBReference** shall contain the parameter that identifies the attribute
20 **GoCBRef** of the **GoCB** for which **MemberReferences** are returned.

21 **15.2.2.3.3.2 ConfigurationRevision**

22 The parameter **ConfigurationRevision** shall contain the attribute **ConfRev** of the **GoCB**.

23 **15.2.2.3.3.3 DataSet**

24 The parameter **DatSet** shall contain the attribute **DatSet** of the **GoCB**.

1 15.2.2.3.3.4 MemberReference [1..n]

2 The parameter **MemberReference** shall contain the **MemberReference** requested for the
3 **MemberOffset** of a member of the **DATA-SET**. A value of NULL shall indicate that no mem-
4 ber of the referenced **DATA-SET** is defined for the member being requested with
5 a **MemberOffset**.

6 15.2.2.3.4 Response–

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

9 15.2.2.4 GetGOOSEElementNumber

10 15.2.2.4.1 GetGOOSEElementNumber parameter table

11 A client shall use the **GetGOOSEElementNumber** service to retrieve the member position of
12 a selected **DataAttribute** in the **DATA-SET** associated with a **GoCB**.

Parameter name
Request
GoCBReference
MemberReference [1..n]
Response+
GoCBReference
ConfigurationRevision
DatSet
MemberOffset [1..n]
Response–
ServiceError

13

14 15.2.2.4.2 Request

15 15.2.2.4.2.1 GoCBReference

16 The parameter **GoCBReference** shall identify the attribute **GoCBRef** of the **GoCB** for which
17 **MemberOffset** are being requested.

18 15.2.2.4.2.2 MemberReference [1..n]

19 The parameter **MemberReference** shall contain the **MemberReference** for which the **Mem-**
20 **berOffset** of a member of the **DATA-SET** is requested. A value of NULL is reserved to indi-
21 cate that no member of the referenced **DATA-SET** is defined for the member being requested
22 with a **MemberReference**.

23 15.2.2.4.3 Response+

24 15.2.2.4.3.1 GoCBReference

25 The parameter **GoCBReference** shall contain the parameter that identifies the attribute
26 **GoCBRef** of the **GoCB** for which **MemberOffsets** are returned.

27 15.2.2.4.3.2 ConfigurationRevision

28 The parameter **ConfigurationRevision** shall contain the attribute **ConfRev** of the **GoCB**.

1 **15.2.2.4.3.3 DataSet**

2 The parameter **DataSet** shall contain the attribute **DataSet** of the **GoCB**.

3 **15.2.2.4.3.4 MemberOffset [1..n]**

4 The parameter **MemberOffset** shall contain the **MemberOffset** requested for the **Member-**
5 **Reference** of a member of the **DATA-SET**. A value of NULL shall indicate that no member of
6 the referenced **DATA-SET** is defined matching with a **MemberReference**.

7 **15.2.2.4.4 Response–**

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

10 **15.2.2.5 GetGoCBValues**

11 A client shall use the **GetGoCBValues** service to retrieve attribute values of **GoCB** made visi-
12 ble and thus accessible to the requesting client by the referenced **LLN0**.

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

Parameter name
Request
GoCBReference
FunctionalConstraint
Response+
GoEnable
GOOSEID
DataSetReference
ConfigurationRevision
NeedsCommissioning
Response–
ServiceError

15

16 **15.2.2.5.1 Request**

17 **15.2.2.5.1.1 GoCBReference**

18 The parameter **GoCBReference** shall specify the **ObjectReference** of the **GoCB**.

19 The service parameter **LCBReference** shall be **LDName/LLN0.GoCBName**.

20 **15.2.2.5.1.2 FunctionalConstraint**

21 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
22 rameter to filter the respective instances of attributes of a **GoCB**.

23 The service parameter **FunctionalConstraint** shall be **GO** (goose control).

24 **15.2.2.5.2 Response+**

25 The parameter **Response+** shall indicate that the service request succeeded.

1 **15.2.2.5.2.1 GoEnable**

2 The parameter **GoEnable** shall contain the value of the corresponding attribute **GoEna** of the
3 referenced **GoCB**.

4 **15.2.2.5.2.2 GOOSEID**

5 The parameter **GOOSEID** shall contain the value of the corresponding attribute **GoID** of the
6 referenced **GoCB**.

7 **15.2.2.5.2.3 DataSetReference**

8 The parameter **DataSetReference** shall contain the value of the corresponding attribute
9 **DatSet** of the referenced **GoCB**.

10 **15.2.2.5.2.4 ConfigurationRevision**

11 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute
12 **ConfRev** of the **GoCB**.

13 **15.2.2.5.2.5 NeedsCommissioning**

14 The parameter **NeedsCommissioning** shall contain the value of the corresponding attribute
15 **NdsCom** of the **GoCB**.

16 **15.2.2.5.3 Response–**

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

19 **15.2.2.6 SetGoCBValues**

20 A client shall use the **SetGoCBValues** service to set attribute values of **GoCB** made visible
21 and thus accessible to the requesting client by the referenced **LLNO**.

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

24

Parameter name
Request
GoCBReference
FunctionalConstraint
GoEnable [0..1]
GOOSEID [0..1]
DataSetReference [0..1]
Response+
Response–
ServiceError

25

26 **15.2.2.6.1 Request**

27 **15.2.2.6.1.1 GoCBReference**

28 The parameter **GoCBReference** shall specify the **ObjectReference** of the **GoCB**.

1 The service parameter **GoCBReference** shall be **LDName/LLN0.GoCBName**.

2 **15.2.2.6.1.2 FunctionalConstraint**

3 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
4 rameter to filter the respective instances of attributes of a **GoCB**.

5 The service parameter **FunctionalConstraint** shall be **GO** (goose control).

6 **15.2.2.6.1.3 GoEnable [0..1]**

7 The parameter **GoEnable** shall contain the value for the corresponding attribute **GoEna** of the
8 referenced **GoCB**.

9 **15.2.2.6.1.4 GOOSEID [0..1]**

10 The parameter **GOOSEID** shall contain the value for the corresponding attribute **GoID** of the
11 referenced **GoCB**.

12 **15.2.2.6.1.5 DataSetReference [0..1]**

13 The parameter **DataSetReference** shall contain the value for the corresponding attribute
14 **DatSet** of the referenced **GoCB**.

15 **15.2.2.6.2 Response+**

16 The parameter **Response+** shall indicate that the service request succeeded.

17 **15.2.2.6.3 Response–**

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

20 This service shall return a failure if the service has been issued for any attribute of a **GoCB**
21 other than **GoEnable** while **GoCB** is enabled.

22 **15.2.3 Generic object oriented substation event (GOOSE) message**

23 **15.2.3.1 GOOSE message syntax**

24 The abstract **GOOSE** message format shall specify the information to be included in the
25 **GOOSE** message. The structure of the **GOOSE** message shall be as specified in Table 30.

26 A GOOSE message shall at least be sent each time when a value from one or more members
27 referenced by the **DATA-SET** change.

1

Table 30 – GOOSE message definition

GOOSE message		
Parameter name	Parameter type	Value/value range/explanation
DatSet	ObjectReference	Value from the instance of GoCB
GoID	VISIBLE STRING65	Value from the instance of GoCB
GoCBRef	ObjectReference	Value from the instance of GoCB
T	TimeStamp	
StNum	INT32U	
SqNum	INT32U	
Test	BOOLEAN	(TRUE) test (FALSE) no-test
ConfRev	INT32U	Value from the instance of GoCB
NdsCom	BOOLEAN	Value from the instance of GoCB
GOOSEData [1..n]		
Value	(*)	(*) type depends on the common data classes defined in IEC 61850-7-3. The parameter shall be derived from GOOSE control

2

3 **15.2.3.2 DatSet – data set**

4 The parameter **DatSet** shall contain the **ObjectReference** of the **DATA-SET** (taken from the
5 **GoCB**) whose values of the members shall be transmitted.

6 **15.2.3.3 GoID – application identifier**

7 The parameter **GoID** shall contain the value of the attribute GoID of the GoCB.

8 **15.2.3.4 GoCBRef – GOOSE control block reference**

9 The parameter **GoCBRef** shall contain the reference of the **GOOSE** control block.

10 **15.2.3.5 T – time stamp**

11 The parameter **T** shall contain the time at which the attribute **StNum** was incremented.

12 **15.2.3.6 StNum – state number**

13 The parameter **StNum** shall contain the counter that increments each time a **GOOSE** message
14 has been sent and a value change has been detected within the **DATA-SET** specified by
15 **DatSet**.

16 The initial value for **StNum** upon a transition of GoEna to TRUE shall be 1. The value of 0 shall
17 be reserved.

18 NOTE – a transition of GoEna to TRUE happens directly at power-up or restart for properly and consistently config-
19 ured GoCB.

20 **15.2.3.7 SqNum – sequence number**

21 The parameter **SqNum** shall contain the counter that shall increment each time a **GOOSE**
22 message has been sent.

1 Following a **StNum** change, the counter **SqNum** shall be set to a value of 0. If the counter
2 **SqNum** overruns, it shall be set to a value of 1. The initial value for **SqNum** upon a transition
3 of GoEna to TRUE shall be 1.

4 **15.2.3.8 Test – test**

5 The parameter **Test** shall indicate with the value of TRUE that the values of the message shall
6 not be used for operational purposes.

7 **15.2.3.9 ConfRev – configuration revision**

8 The parameter **ConfRev** (taken from the **GoCB**) shall contain the count of the number of times
9 that the configuration of the **DATA-SET** referenced by **DatSet** has been changed.

10 **15.2.3.10 NdsCom – needs commissioning**

11 The parameter **NdsCom** shall contain the attribute **NdsCom** (taken from the **GoCB**) of the
12 **GoCB**.

13 **15.2.3.11 GOOSEData [1..n]**

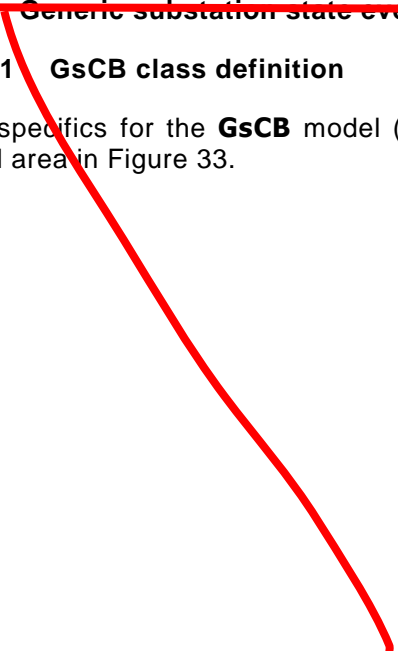
14 The parameter **GOOSEData** shall contain the user-defined information (of the members of
15 **DATA-SET**) to be included in a **GOOSE** message.

16 The parameter **Value** shall contain the value of a member of the **DATA-SET** referenced in the
17 **GoCB**.

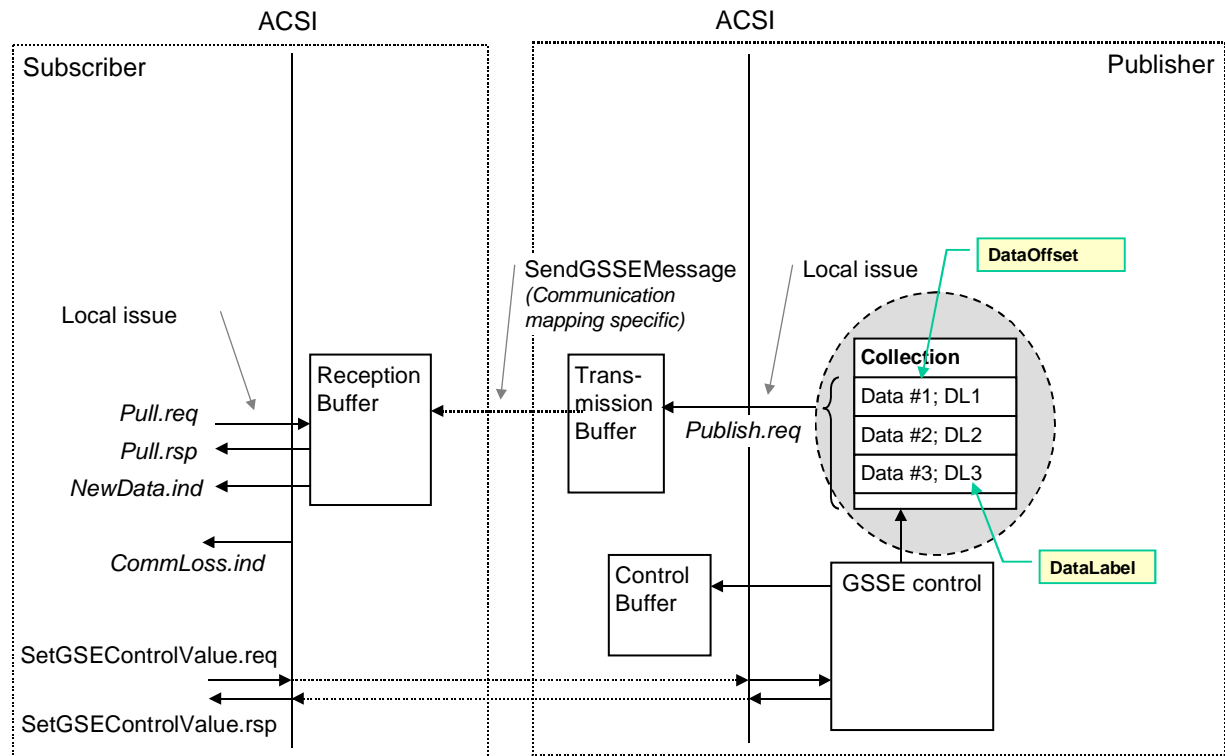
18 ~~15.3 Generic substation state event (GOOSE) control block (GsCB)~~

19 **15.3.1 GsCB class definition**

20 The specifics for the **GsCB** model (compared to the **GoCB** model) are depicted in the shad-
21 owed area in Figure 33.



Move to informative
Annex



IEC 422/03

Figure 33 – Specifics for GsCB model

The information to be sent shall be a **Collection** of data. The data shall be uniquely numbered from 1 to higher numbers. Each data shall have a **DataLabel**.

The **GsCB** shall be as defined in Table 31.

Table 31 – GSSE control block class definition

GsCB class			
Attribute name	Attribute type	FC	Value/value range/explanation
GsCBName	ObjectName		Instance name of an instance of GsCB
GsCBRef	ObjectReference		Path-name of an instance of GsCB
GsEna	BOOLEAN	GS	Enabled (TRUE) disabled (FALSE)
GsID	VISIBLE STRING65	GS	
DataLabel [1..n]	VISIBLE STRING65	GS	
LSentData [1..n]	GSSEData	GS	Derived from GSSE message
DstAddress	PHYCOMADDR	GS	
Services			
SendGSSEMessage			
GetGsReference			
GetGSSEDataOffset			
GetGsCBValues			
SetGsCBValues			

1 **15.3.2 Generic substation state event (GSSE) control block class attributes**

2 **15.3.2.1 GsCBName – GSSE control name**

3 The attribute **GsCBName** shall unambiguously identify a **GsCB** within the scope of a **LLN0**.

4 **15.3.2.2 GsCBRef – GSSE control reference**

5 The attribute **GsCBRef** shall be the unique path-name of a **GsCB** within a **LLN0**.

6 The ObjectReference **GsCBRef** shall be:

LDName/LLN0.GsCBName

7 **15.3.2.3 GsEna – GSSE enable**

8 The attribute **GsEna** (if set to TRUE) shall indicate that **GsCB** is currently enabled to send val-
9 ues of the **GsCB**. If set to FALSE the **GsCB** shall stop sending **GSSE** messages.

10 While being TRUE (**GsCB** enabled), no changes of attribute values of the **GsCB** other than dis-
11 abling shall be allowed.

12 If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client that has enabled the **GsCB** is
13 lost, the instance of **GsCB** shall set the attribute to FALSE.

14 **15.3.2.4 GsID – GSSE identification**

15 The attribute **GsID** shall be a user definable unique identification of the context of the GSSE
16 message.

17 NOTE 1 The context of the GSSE message is defined by the values configured in the GsCB.

18 NOTE 2 Depending upon the SCSM and actual implementation, it may not be possible to uniquely identify the
19 GSSE control through the control reference. Therefore, a standardized control attribute must be provided to allow
20 the system configuration process to be able to uniquely identify the control within the scope of the substation.

21 **15.3.2.5 DataLabel [1..n]**

22 The attribute **DataLabel** of visible strings shall contain a reference for each entry used within
23 the attribute **LastSentData**. A NULL value shall indicate that that particular **LastSentData**
24 data entry is not in use. The DEFAULT value is a local issue.

25 The visible string shall hold the value of the **ObjectReference** if the corresponding element is
26 being sent. Otherwise the value of the **ObjectReference** shall be NULL. The DEFAULT value
27 shall be **GsCBName**.

28 NOTE The attribute **DataLabel** allows a user to assign a system unique identifier for the application that is issu-
29 ing the **GSSE**.

30 **15.3.2.6 LSentData [1..n] – last sent data values**

31 The attribute **LSentData** shall represent the data values that have been sent with the last
32 **GSSE** message.

33 The maximum for the number of data values shall be at least 24; i.e. the attribute **LSentData**
34 shall be capable of holding at least 24 double-bit status values.

35 NOTE The maximum number of data values may be constrained by the SCSM and local means.

1 15.3.2.7 DstAddress

2 The attribute DstAddress shall be the SCSM specific addressing information like media access
3 address, priority, and other information.

4 ~~15.3.3 GSSE service definitions~~

GSSE move to an informative annex ...
Zug, 2007-07-11

5 15.3.3.1 Overview

6 For the **GsCB** the following services are defined:

Service	Description
SendGSSEMessage	Send GSSE message
GetGsReference	Retrieve the DataLabel of a specific value associated with the GSSE message
GetGSSEDataOffset	Retrieve the position of the specific value associated with the GSSE message of a DataLabel
GetGsCBValues	Retrieve the attributes of a GsCB
SetGsCBValues	Write the attributes of a GsCB

7

8 15.3.3.2 SendGSSEMessage

9 15.3.3.2.1 SendGSSEMessage parameter table

10 The **SendGSSEMessage** service shall be used by a **GsCB** to send a **GSSE** message over a
11 **MULTICAST-APPLICATION-ASSOCIATION**.

Parameter name
Request
GSSE message

12

13 15.3.3.2.2 Request

14 15.3.3.2.2.1 GSSE message

15 The parameter **GSSE message** shall specify the **GSSE** message as defined in 15.3.4 of the
16 given **GsCB**.

17 15.3.3.3 GetGsReference

18 15.3.3.3.1 GetGsReference parameter table

19 A client shall use the **GetGsReference** service to retrieve the **DataLabels** of specific mem-
20 bers of the **Collection** of the referenced **GsCB**.

Parameter name
Request
GsCBReference
DataOffset [1..n]
Response+
GsCBReference
DataLabel [1..n]
Response-

ServiceError

1

2 **15.3.3.3.2 Request**3 **15.3.3.3.2.1 GsCBReference**

4 The parameter **GsCBReference** shall identify the attribute **GsCBRef** of the **GsCB** for which
5 **DataLabels** are being requested.

6 **15.3.3.3.2.2 DataOffset [1..n]**

7 The parameter **DataOffset** shall contain a number identifying a member of the **Collection**.

8 **15.3.3.3.3 Response+**9 **15.3.3.3.3.1 GsCBReference**

10 The parameter **GsCBReference** shall contain the parameter that identifies the attribute
11 **GoCBRef** of the **GsCB** for which **DataLabels** are returned.

12 **15.3.3.3.3.2 DataLabel [1..n]**

13 The parameter **DataLabel** shall contain the **DataLabel** requested for the **DataOffset** of the
14 **Collection**. A value of NULL shall indicate that no member is defined for the member being
15 requested with the respective **DataOffset**.

16 **15.3.3.3.4 Response–**

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

19 **15.3.3.4 GetGSSEDataOffset**20 **15.3.3.4.1 GetGSSEDataOffset parameter table**

21 A client shall use the **GetGSSEDataOffset** service to retrieve the data position of a selected
22 data in the **Collection** associated with a **GsCB**.

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

25

Parameter name
Request
GsCBReference
DataLabel [1..n]
Response+
GsCBReference
DataOffset [1..n]
Response–
ServiceError

26

1 **15.3.3.4.2 Request**2 **15.3.3.4.2.1 GsCBReference**

3 The parameter **GsCBReference** shall identify the attribute **GsCBRef** of the **GsCB** for which
4 **MemberOffset** are being requested.

5 **15.3.3.4.2.2 DataLabel [1..n]**

6 The parameter **DataLabel** shall contain the **DataLabel** for which the **DataOffset** of the **Col-**
7 **lection** is requested.

8 **15.3.3.4.3 Response+**9 **15.3.3.4.3.1 GsCBReference**

10 The parameter **GsCBReference** shall contain the parameter that identifies the attribute
11 **GoCBRef** of the **GsCB** for which **DataLabels** are returned.

12 **15.3.3.4.3.2 DataOffset [1..n]**

13 The parameter **DataOffset** shall contain a number identifying a member of the **Collection**.
14 A value of NULL shall indicate that no **DataOffset** is defined for the member being requested
15 with the respective **DataLabel**.

16 **15.3.3.4.4 Response–**

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

19 **15.3.3.5 GetGsCBValues**

20 A client shall use the **GetGsCBValues** service to retrieve attribute values of **GsCB** made visi-
21 ble and thus accessible to the requesting client by the referenced **LLNO**.

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

24

Parameter name
Request
GsCBReference
FunctionalConstraint
Response+
GsEnable
GSSEID
DataLabel [1..n]
LastSentData [1..n]
Response–
ServiceError

25

1 **15.3.3.5.1 Request**2 **15.3.3.5.1.1 GsCBReference**3 The parameter **GsCBReference** shall specify the **ObjectReference** of the **GsCB**.4 The service parameter **GsCBReference** shall be **LDName/LLN0.GsCBName**.5 **15.3.3.5.1.2 FunctionalConstraint**6 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
7 rameter to filter the respective instances of attributes of a **GsCB**.8 The service parameter **FunctionalConstraint** shall be **GS** (gsse control).9 **15.3.3.5.2 Response+**10 The parameter **Response+** shall indicate that the service request succeeded.11 **15.3.3.5.2.1 GsEnable**12 The parameter **GsEnable** shall contain the value of the corresponding attribute **GsEna** of the
13 referenced **GsCB**.14 **15.3.3.5.2.2 GSSEID**15 The parameter **GSSEID** shall contain the value of the corresponding attribute **GsID** of the ref-
16 erenced **GsCB**.17 **15.3.3.5.2.3 DataLabel [1..n]**18 The parameter **DataLabel** shall contain the **DataLabel** of the **Collection**.19 **15.3.3.5.2.4 LastSentData [1..n]**20 The parameter **LastSentData** shall contain the value of the attribute **LSentData** of the **GsCB**.21 **15.3.3.5.3 Response–**22 The parameter **Response–** shall indicate that the service request failed. The appropriate **Ser-**
23 **viceError** shall be returned.24 **15.3.3.6 SetGsCBValues**25 A client shall use the **SetGsCBValues** service to set attribute values of **GsCB** made visible and
26 thus accessible to the requesting client by the referenced **LLNO**.27 NOTE The visible instances are those that are defined within a given view (see Clause 7 for details on
28 the view concept).

29

Parameter name
Request
GsCBReference
FunctionalConstraint
GsEnable [0..1]

GSSEID [0..1]
Response+
Response–
ServiceError

1

2 **15.3.3.6.1 Request**3 **15.3.3.6.1.1 GsCBReference**4 The parameter **GsCBReference** shall specify the ObjectReference of the **GsCB**.5 The service parameter **GsCBReference** shall be **LDName/LLN0.GsCBName**.6 **15.3.3.6.1.2 FunctionalConstraint**7 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
8 rameter to filter the respective instances of attributes of a **GsCB**.9 The service parameter **FunctionalConstraint** shall be **GS** (gsse control).10 **15.3.3.6.1.3 GsEnable [0..1]**11 The parameter **GsEnable** shall contain the value for the corresponding attribute **GsEna** of the
12 referenced **GsCB**.13 **15.3.3.6.1.4 GSSEID [0..1]**14 The parameter **GSSEID** shall contain the value for the corresponding attribute **GsID** of the ref-
15 erenced **GsCB**.16 **15.3.3.6.2 Response+**17 The parameter **Response+** shall indicate that the service request succeeded.18 **15.3.3.6.3 Response–**19 The parameter **Response–** shall indicate that the service request failed. The appropriate **Ser-**
20 **viceError** shall be returned.21 This service shall return a failure if the service has been issued for any attribute of a **GsCB**
22 other than **GsEnable** while **GsCB** is enabled.23 **15.3.4 Generic substation state event (GSSE) message**24 **15.3.4.1 Syntax**25 The abstract **GSSE** message format shall specify the information to be included in the **GSSE**
26 message. The structure of the **GSSE** message shall be as specified in Table 32.27 A **GSSE** message shall at least be sent each time when a value from one or more of the
28 **LSentData** change (for example, a change of status value is detected).

1

Table 32 – GSSE message definition

GSSE message		
Parameter name	Parameter type	Value/value range/explanation
GsID	VISIBLE STRING65	Value from the instance of GsCB
T	EntryTime	
SqNum	INT32U	
StNum	INT32U	
Test	BOOLEAN	(TRUE) test (FALSE) no-test
PhsID	INT16U	
GSSEData [1..n]		
Value	CODED ENUM	Invalid or transient (0) false or closed (1) true or open (2) invalid (3)

2

15.3.4.2 GsID – application identifier

The parameter **GsID** shall contain the value of the attribute GsID of the GsCB.

15.3.4.3 T – time stamp

The parameter **T** shall contain the time at which the **StNum** attribute was incremented.

15.3.4.4 SqNum – sequence number

The parameter **SqNum** shall contain the counter that shall increment each time a GSSE message has been sent.

The initial value for **STNum** shall be 1. The value of 0 shall be reserved.

15.3.4.5 StNum – state number

The parameter **StNum** shall contain a counter that increments each time a GSSE message has been sent and a value change has been detected within the data values of **LSentData**.

The initial value for **StNum** shall be 1. The value of 0 shall be reserved.

15.3.4.6 Test – test

The parameter **Test** shall indicate with the value of TRUE that the values of the message shall not be used for operational purposes.

15.3.4.7 PhsID – phase identification

The parameter PhsID shall indicate faulted phases.

15.3.4.8 GSSEData [1..n]

The parameter **GSSEData** shall be a status value of 4 values coded as CODED ENUM. The defined values are invalid or transient (0) | false or closed (1) | true or open (2) | invalid (3).

The size of the array [1..n] is determined by the size of the **LSentData** attribute of the associated **GsCB**.

1 16 Transmission of sampled value class model

2 16.1 Overview

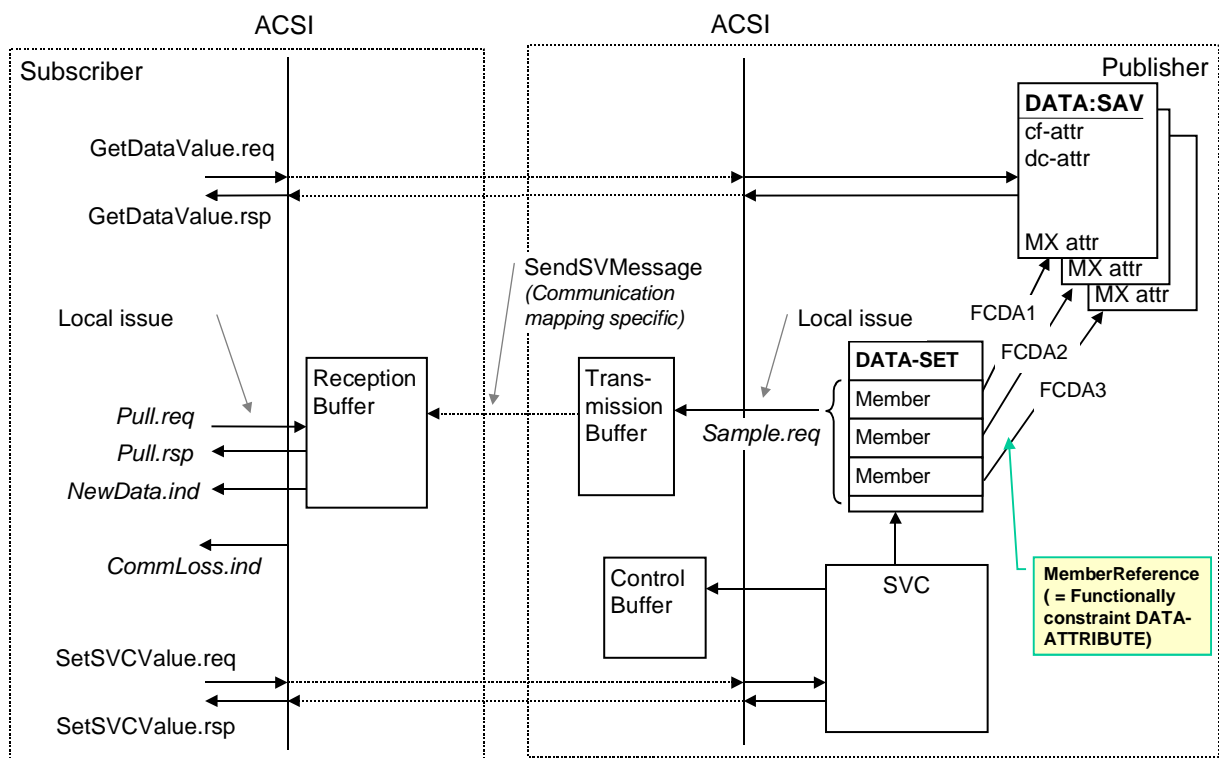
3 The transmission of sampled values requires special attention with regard to the time con-
4 straints. The model provides transmission of sampled values in an organized and time- con-
5 trolled way so that the combined jitter of sampling and transmission is minimized to a degree
6 that an unambiguous allocation of the samples, times, and sequence is provided.

7 The model applies to the exchange of values of a **DATA-SET**. The **DATA** of the **DATA-SET** are
8 of the common data class SAV (sampled value as defined in IEC 61850-7-3). A buffer structure
9 shall be defined for the transmission of the sampled values.

10 The information exchange shall be based on a publisher/subscriber mechanism. The publisher
11 shall write the values in a local buffer at the sending side; the subscriber shall read the values
12 from a local buffer at the receiving side. A time stamp shall be added to the values, so that the
13 subscriber can check the timeliness of the values. The communication system shall be respon-
14 sible to update the local buffers of the subscribers. A sampled value control (**SVC**) in the pub-
15 lisher shall be used to control the communication procedure.

16 Figure 34 gives an overview on the classes and services of the model.

17



IEC 423/03

18
19
20 **Figure 34 – Model for transmission of sampled values**

21 There shall be two methods to exchange sampled values between a publisher and one or more
22 subscriber. One method shall use the **MULTICAST-APPLICATION-ASSOCIATION** (multicast
23 sampled value control, **MSVCB**), the other method shall use the **TWO-PARTY-**
24 **APPLICATION-ASSOCIATION** (unicast sampled value control, **USVCB**).

25 The producer shall sample the inputs with the specified sample rate. The synchronization of
26 this sampling may be done internal or over the network. The samples shall be posted in the
27 transmission buffer.

1 The network embedded scheduler shall send the content of the buffer over the network to the
 2 subscribers. The rate may be a mapping specific parameter. Then the samples shall be placed
 3 into the receive buffers of the subscribers. The arrival of a new series of samples in the receive
 4 buffer shall be signalled to the application.

5 The model shall provide mechanisms that the subscriber can detect lost samples. If samples
 6 are not be transmitted due to problems in the communication network, the publisher shall de-
 7 delete these samples.

8 **16.2 Transmission of sampled values using multicast**

9 The transmission of sampled values using multicast (**MULTICAST-SAMPLE-VALUE-**
 10 **CONTROL-BLOCK – MSVCB**) shall be based on configured configuration in the producer de-
 11 vice. The data exchange shall be based on the multicast application association. To support
 12 self-descriptive capabilities, any client may read the attributes of the sampled value control in-
 13 stance. Authorized clients may modify attributes of the sampled value control.

14 **16.2.1 MSVCB class definition**

15 The **MSVCB** shall be as defined in Table 33.

16 **Table 33 – MSVCB class definition**

MSVCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
MsvCBNam	ObjectName	-	-	Instance name of an instance of MSVCB
MsvCBRef	ObjectReference	-	-	Path-name of an instance of MSVCB
SvEna	BOOLEAN	MS	dchg	Enabled (TRUE) disabled (FALSE), DEFAULT FALSE
MsvID	VISIBLE STRING65	MS	-	
DatSet	ObjectReference	-	-	
ConfRev	INT16U	-	-	
SmpRate	INT16U	MS	-	(0..MAX)
OptFlds	PACKED LIST	MS	dchg	
refresh-time	BOOLEAN			
sample-synchronized	BOOLEAN			
sample-rate	BOOLEAN			
data-set-name	BOOLEAN			
Services				
SendMSVMessage				
GetMSVCBValues				
SetMSVCBValues				

Value is split: See new table in 7-3.

17 **16.2.1.1 MsvCBNam – multicast sampled value control name**

18 The attribute **MsvCBNam** shall unambiguously identify a **MSVCB** within the scope of an **LLNO**.

19 **16.2.1.2 MsvCBRef – multicast sampled value control reference**

20 The attribute **MsvCBRef** shall be the unique path-name of a **MSVCB** within an **LLNO**.

21 The **ObjectReference MsvCBRef** shall be:

LDName/LLN0.MsvCBNam

1

2 **16.2.1.3 SvEna – sampled value enable**

3 The attribute **SvEna** (if set to TRUE) shall indicate that the **MSVCB** is currently enabled to send
4 values of the **MSVCB**. If set to FALSE the **MSVCB** shall stop sending values.

5 While being TRUE (**MSVCB** enabled), no changes of attribute values of the **MSVCB** other than
6 disabling shall be allowed.

7 **16.2.1.4 MsvID – multicast sampled value identifier**

8 The attribute **MSVID** shall be a unique identification of the sampled value buffer related to the
9 update of the sampled values.

10 **16.2.1.5 DatSet**

11 The attribute **DatSet** shall specify the reference of the **DATA-SET** whose values of members
12 are to be transmitted in the **MSVCB** message.

13 **16.2.1.6 ConfRev – configuration revision**

14 The attribute **ConfRev** shall contain a count of the number of times that the configuration with
15 regard to the **MSVCB** has been changed. Changes that shall be counted are:

- 16 – any deletion of a member of the **DATA-SET**,
- 17 – any reordering of members of the **DATA-SET**,
- 18 – any change of a value of the **DataAttribute** of the **MSVCB** (functional constraint
19 equals CF,
- 20 – any change of a value of **an attribute** of **MSVCB** (functional constraint of attribute **MSVCB**)
21 equals **MS** (multicast sampled value control).

22 The counter shall be incremented when the configuration changes.

23 The initial value for **ConfRev** is outside the scope of this standard. The value of 0 shall be re-
24 served. A restart of the IED shall not reset the value.

25 NOTE Configuration changes of **DATA-SETS** due to processing of services are not allowed (see **DATA-SET**
26 model). Changes to be taken into account for the Configuration.
27

28 **16.2.1.7 SmpRate**

29 The attribute **SmpRate** shall specify the sample rate in units of samples per nominal period.

30 **16.2.1.8 OptFlds – optional fields to include in SV message**

31 The attribute **OptFlds** shall be the client-specified optional fields to be included in the **SV** mes-
32 saged by this **MSVCB**. This attribute defines a subset of the optional header fields that
33 shall be included in the **SV** message:

- 34 – **RefrTm** (Refresh time, time of refresh activity)
- 35 – **SmpSynch** (Samples synchronized, samples are synchronized by clock signals) , and
- 36 – **SmpRate** (sample rate from the instance of MSVCB)
- 37 – **DatSet** (data set name)

Is this for any or for
some specific attributes
true.

The model for the sample rate shall be extended in a way, that as well sampling rates of less than one sample per period are possible. This is a requirement from the transmission of phasors (used between substations).

1 16.2.2 Multicast sampled value class services

2 16.2.2.1 Overview

3 For the **MSVCB** the following services are defined:

Service	Description
SendMSVMessage	Send MSV message
GetMSVCBValues	Retrieve the attributes of an MSVCB
SetMSVCBValues	Write the attributes of an MSVCB

4

5 16.2.2.2 SendMSVMessage

6 16.2.2.2.1 SendMSVMessage parameter table

7 The **SendMSVMessage** service shall be used by an **MSVCB** to send sampled values from the
8 server to the client over a **MULTICAST-APPLICATION-ASSOCIATION**.

Parameter name
Request
MSV message

9

10 16.2.2.2.2 Request

11 16.2.2.2.2.1 MSV message

12 The parameter **MSV message** shall specify the values of the members of the referenced
13 **DATA-SET** of the **MSVCB** as specified in the abstract sampled value format definition (see
14 16.4). The concrete format of the **MSV message** shall be defined in the SCSM.

15 16.2.2.3 GetMSVCBValues

16 A client shall use the **GetMSVCBValues** service to retrieve attribute values of **MSVCB** made
17 visible and thus accessible to the requesting client by the referenced **LLNO**.

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

Parameter name
Request
MsvCBReference
FunctionalConstraint
Response+
SvEnable
MulticastSampleValueID
DataSetReference
ConfigurationRevision
SampleRate
OptFlds
Response–
ServiceError

20

1 **16.2.2.3.1 Request**

2 **16.2.2.3.1.1 MsvCBReference**

3 The parameter **MsvCBReference** shall specify the **ObjectReference** of the **MSVCB**.

4 The service parameter **MsvCBReference** shall be **LDName/LLN0.MsvCBNam**.

5 **16.2.2.3.1.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
7 rameter to filter the respective instances of attributes of a **MSVCB**.

8 The service parameter **FunctionalConstraint** shall be **MS** (multicast sampled value control).

9 **16.2.2.3.2 Response+**

10 The parameter **Response+** shall indicate that the service request succeeded.

11 **16.2.2.3.2.1 SvEnable**

12 The parameter **SvEnable** shall contain the value of the corresponding attribute **SvEna** of the
13 referenced **MSVCB**.

14 **16.2.2.3.2.2 MulticastSampleValueID**

15 The parameter **MulticastSampleValueID** shall contain the value of the corresponding attrib-
16 ute **MsvID** of the referenced **MSVCB**.

17 **16.2.2.3.2.3 DataSetReference**

18 The parameter **DataSetReference** shall contain the value of the corresponding attribute
19 **DatSet** of the referenced **MSVCB**.

20 **16.2.2.3.2.4 ConfigurationRevision**

21 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute
22 **ConfRev** of the **MSVCB**.

23 **16.2.2.3.2.5 SampleRate**

24 The parameter **SampleRate** shall contain the value of the corresponding attribute **SmpRate** of
25 the **MSVCB**.

26 **16.2.2.3.2.6 OptionalFields**

27 The parameter **OptFlds** shall contain the value of the corresponding attribute **OptFlds** of the
28 **MSVCB**.

29 **16.2.2.3.3 Response–**

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

32 **16.2.2.4 SetMSVCBValues**

33 A client shall use the **SetMSVCB Values** service to set attribute values of **MSVCB** made visi-
34 ble and thus accessible to the requesting client by the referenced **LLNO**.

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

1

Parameter name
Request
MsvCBReference
FunctionalConstraint
SvEnable [0..1]
MulticastSampleValueID [0..1]
DataSetReference [0..1]
SampleRate [0..1]
OptFlds [0..1]
Response+
Response-
ServiceError

2

3 **16.2.2.4.1 Request**4 **16.2.2.4.1.1 MsvCBReference**5 The parameter **MsvCBReference** shall specify the **ObjectReference** of the **MSVCB**.6 The service parameter **MsvCBReference** shall be **LDName/LLN0.MsvCBNam**.7 **16.2.2.4.1.2 FunctionalConstraint**8 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
9 rameter to filter the respective instances of attributes of a **MSVCB**.10 The service parameter **FunctionalConstraint** shall be **MS** (multicast sampled value control).11 **16.2.2.4.1.3 SvEnable [0..1]**12 The parameter **SvEnable** shall contain the value for the corresponding attribute **SvEna** of the
13 referenced **MSVCB**.14 **16.2.2.4.1.4 MulticastSampleValueID [0..1]**15 The parameter **MulticastSampleValueID** shall contain the value for the corresponding attrib-
16 ute **MsvID** of the referenced **MSVCB**.17 **16.2.2.4.1.5 DataSetReference [0..1]**18 The parameter **DataSetReference** shall contain the value for the corresponding attribute
19 **DatSet** of the referenced **MSVCB**.20 **16.2.2.4.1.6 SampleRate [0..1]**21 The parameter **SampleRate** shall contain the value for the corresponding attribute **SmpRate**
22 of the **MSVCB**.23 **16.2.2.4.1.7 OptionalFields**24 The parameter **OptFlds** shall contain the value of the corresponding attribute **OptFlds** of the
25 **MSVCB**.

1 16.2.2.4.2 Response+

2 The parameter **Response+** shall indicate that the service request succeeded.

3 16.2.2.4.3 Response–

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

6 This service shall return a failure if the service has been issued for any attribute of a **MSVCB** other than **SvEnable** while **MSVCB** is enabled.

8 16.3 Transmission of sampled values using unicast

9 The transmission of sampled values using unicast (**UNICAST-SAMPLE-VALUE-CONTROL-BLOCK – USVCB**) shall be based on two-party application associations. The subscriber shall establish the association with the producer. The subscriber may then configure the class and enable the transmission of the sampled values with the attribute **SvEna**. When the association is released, the transmission of the sampled values shall stop and the instance of the control class shall be released.

15 The samples shall be sent using the two-party application association.

16 16.3.1 USVCB class definition

17 The **USVCB** shall be as defined in Table 34.

18 **Table 34 – USVCB class definition**

USVCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
UsvCBNam	ObjectName	-	-	Instance name of an instance of UNICAST-SVC
UsvCBRef	ObjectReference	-	-	Path-name of an instance of UNICAST-SVC
SvEna	BOOLEAN	US	dchg	Enabled (TRUE) disabled (FALSE), DEFAULT FALSE
Resv	BOOLEAN	US	-	
UsvID	VISIBLE STRING65	US	-	
DatSet	ObjectReference	US	dchg	
ConfRev		US	dchg	
SmpRate		US	dchg	(0..MAX)
OptFlds	PACKED LIST	US	dchg	
refresh-time	BOOLEAN			
sample-synchronized	BOOLEAN			
sample-rate	BOOLEAN			
data-set-name	BOOLEAN			
Services				
SendUSVMessage				
GetUSVCBValues				
SetUSVCBValues				

19

20 16.3.1.1 UsvCBNam – unicast sampled value control name

21 The attribute **UsvCBNam** shall unambiguously identify a **USVCB** within the scope of a **LLNO**.

1 **16.3.1.2 UsvCBRef – unicast sampled value control reference**

2 The attribute **UsvCBRef** shall be the unique path-name of a **USVCB** within a **LLNO**.

3 The **ObjectReference UsvCBRef** shall be:

LDName/LLNO.UsvCBNam

4

5 **16.3.1.3 SvEna – sampled value enable**

6 The attribute **SvEna** (if set to TRUE) shall indicate that the **USVCB** is currently enabled to send
7 values of the **USVCB**. If set to FALSE the **USVCB** shall stop issuing reports.

8 While being TRUE (**USVCB** enabled), no changes of attribute values of **USVCB** other than dis-
9 abling shall be allowed.

10 If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client that has enabled the **USVCB**
11 is lost, the **USVCB** shall set the attribute to FALSE.

12 **16.3.1.4 Resv – reserve USVCB**

13 The attribute **Resv** (if set to TRUE) shall indicate that the **USVCB** is currently exclusively re-
14 served for the client that has set the value to TRUE. Other clients shall not be allowed to a set
15 any attribute of that **USVCB**.

16 If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client that has set this attribute to
17 TRUE is lost, the **USVCB** shall set the attribute to FALSE.

18 NOTE The attribute **Resv** functions as a semaphore for the configuration, enabling and disabling of the **USVCB**.

19 **16.3.1.5 UsvID**

20 The attribute **UsvID** shall be a unique identification of the sampled value buffer related to the
21 update of the sampled values.

22 **16.3.1.6 DatSet**

DatSet
Bergen

23 The attribute ~~**DatRef**~~ shall specify the reference of the **DATA-SET** whose values of members
24 are to be transmitted in the **USVCB** message.

25 **16.3.1.7 ConfRev – configuration revision**

26 The attribute **ConfRev** shall contain a count of the number of times that the configuration with
27 regard to the **USVCB** has been changed. Changes that shall be counted are:

- 28 – any deletion of a member of the **DATA-SET**,
- 29 – any reordering of members of the **DATA-SET**,
- 30 – any change of a value of the **DataAttribute** of the **DATA-SET** whose functional constraint
31 equals cf.
- 32 – any change of a value of an attribute of **USVCB** (functional constraint of attribute **USVCB**
33 equals **US**).

34 The counter shall be incremented when the configuration changes.

1 The initial value for **ConfRev** is outside the scope of this standard. The value of 0 shall be re-
 2 served. A restart of the IED shall not reset the value.

3 NOTE Configuration changes of **DATA-SETs** due to production changes to be taken into account for the ConfRev are thus
 4

The model for the sample rate shall be extended in a way, that as well sampling rates of less than one sample per period are possible. This is a requirement from the transmission of phasors (used between substations).

5 16.3.1.8 SmpRate

6 The attribute **SmpRate** shall specify the sample rate in units of samples per nominal period.

7 16.3.1.9 OptFlds – optional fields to include in SV message

8 The attribute **OptFlds** shall be the client-specified optional fields to be included in the **SV** mes-
 9 sage issued by this **USVCB**. This attribute defines a subset of the optional header fields that
 10 shall be included in the **SV** message:

- 11 – **RefrTm** (Refresh time, time of refresh activity)
- 12 – **SmpSynch** (Samples synchronized, samples are synchronized by clock signals), and
- 13 – **SmpRate** (sample rate from the instance of **USVCB**)
- 14 - **DatSet** (data set name)

15 16.3.2 Unicast sampled value services

16 16.3.2.1 Overview

17 For the **USVCB** the following services are defined:

Service	Description
SendUSVMessage	Send USV message
GetUSVCBValues	Retrieve the attributes of a USVCB
SetUSVCBValues	Write the attributes of a USVCB

18

19 16.3.2.2 SendUSVMessage

20 16.3.2.2.1 SendUSVMessage parameter table

21 The **SendUSVMessage** service shall be used by a **USVCB** to send sampled values from the
 22 server to the client over a **TWO-PARTY-APPLICATION-ASSOCIATION**.

Parameter name
Request
USV message

23

24 16.3.2.2.2 Request

25 USV message

26 The parameter **USV message** shall specify the values of the members of the referenced
 27 **DATA-SET** of the **USVCB** as specified in the abstract sampled value format definition
 28 (see 16.4). The concrete format of the **USV message** shall be defined in the SCSM.

29 16.3.2.3 GetUSVCBValues

30 A client shall use the **GetUSVCBValues** service to retrieve attribute values of **USVCB** made
 31 visible and thus accessible to the requesting client by the referenced **LLNO**.

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

3

Parameter name
Request
UsvCBReference
FunctionalConstraint
Response+
SvEnable
CBReserved
UnicastSampleValueID
DataSetReference
ConfigurationRevision
SampleRate
Response–
ServiceError

4

5 **16.3.2.3.1 Request**

6 **16.3.2.3.1.1 UsvCBReference**

7 The parameter **UsvCBReference** shall specify the **ObjectReference** of the **USVCB**.

8 The service parameter **UsvCBReference** shall be **LDName/LLN0.UsvCBNam**.

9 **16.3.2.3.1.2 FunctionalConstraint**

10 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
11 rameter to filter the respective instances of attributes of a **USVCB**.

12 The service parameter **FunctionalConstraint** shall be **US** (unicast sampled value control).

13 **16.3.2.3.2 Response+**

14 The parameter **Response+** shall indicate that the service request succeeded.

15 **16.3.2.3.2.1 SvEnable**

16 The parameter **SvEnable** shall contain the value of the corresponding attribute **SvEna** of the
17 referenced **USVCB**.

18 **16.3.2.3.2.2 CBReserved**

19 The parameter **CBReserved** shall contain the value of the corresponding attribute **Resv** of the
20 referenced **USVCB**.

21 **16.3.2.3.2.3 UnicastSampleValueID**

22 The parameter **UnicastSampleValueID** shall contain the value of the corresponding attribute
23 **UsvID** of the referenced **USVCB**.

1 **16.3.2.3.2.4 DataSetReference**

2 The parameter **DataSetReference** shall contain the value of the corresponding attribute
3 **DatSet** of the referenced **USVCB**.

4 **16.3.2.3.2.5 ConfigurationRevision**

5 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute
6 **ConfRev** of the **USVCB**.

7 **16.3.2.3.2.6 SampleRate**

8 The parameter **SampleRate** shall contain the value of the corresponding attribute **SmpRate** of
9 the **USVCB**.

10 **16.3.2.3.3 Response–**

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

13 **16.3.2.4 SetUSVCBValues**

14 A client shall use the **SetUSVCBValues** service to set attribute values of **USVCB** made visible
15 and thus accessible to the requesting client by the referenced **LLNO**.

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
UsvCBReference
FunctionalConstraint
SvEnable [0..1]
CBReserved [0..1]
UnicastSampleValueID [0..1]
DataSetReference [0..1]
SampleRate [0..1]
Response+
Response–
ServiceError

18

19 **16.3.2.4.1 Request**

20 **16.3.2.4.1.1 UsvCBReference**

21 The parameter **UsvCBReference** shall specify the **ObjectReference** of the **USVCB**.

22 The service parameter **UsvCBReference** shall be **LDName/LLNO.UsvCBNam**.

23 **16.3.2.4.1.2 FunctionalConstraint**

24 The parameter **FunctionalConstraint** shall contain the value of the functional constraint pa-
25 rameter to filter the respective instances of attributes of a **USVCB**.

1 The service parameter **FunctionalConstraint** shall be **US** (unicast sampled value control).

2 **16.3.2.4.1.3 SvEnable [0..1]**

3 The parameter **SvEnable** shall contain the value for the corresponding attribute **SvEna** of the
4 referenced **USVCB**.

5 **16.3.2.4.1.4 CBReserved**

6 The parameter **CBReserved** shall contain the value for the corresponding attribute **Resv** of the
7 referenced **USVCB**.

8 **16.3.2.4.1.5 UnicastSampleValueID [0..1]**

9 The parameter **UnicastSampleValueID** shall contain the value for the corresponding attribute
10 **UsvID** of the referenced **USVCB**.

11 **16.3.2.4.1.6 DataSetReference [0..1]**

12 The parameter **DataSetReference** shall contain the value for the corresponding attribute
13 **DatSet** of the referenced **USVCB**.

14 **16.3.2.4.1.7 SampleRate [0..1]**

15 The parameter **SampleRate** shall contain the value for the corresponding attribute **SmpRate**
16 of the **USVCB**.

17 **16.3.2.4.2 Response+**

18 The parameter **Response+** shall indicate that the service request succeeded.

19 **16.3.2.4.3 Response–**

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

22 This service shall return a failure if the service has been issued for any attribute of a **USVCB**
23 other than **SvEnable** while **USVCB** is enabled.

24 **16.4 Sampled value format**

25 The abstract sampled value format used for the sampled value message shall be as defined in
26 Table 35.

1

Table 35 – Sampled value (SV) format definition

Sampled value format		
Parameter name	Parameter type	Value/value range/explanation
MsvID or UsvID	VISIBLE STRING65	Value from the MSVCB or USVCB
OptFlds	^a	Optional fields to be included in the SV message
DatSet	ObjectReference	OPTIONAL; value from the MSVCB or USVCB
Sample [1..n]		
Value	(*)	(*) The value of the member of the instance of the DATA-SET . Type of the common data classes is SAV (sampled analogue value) as defined in IEC 61850-7-3
SmpCnt	INT16U	Sample counter
RefrTm	TimeStamp	OPTIONAL; time of refresh activity
ConfRev	INT32U	Configuration revision number from the instance of MSVCB or USVCB
SmpSynch	INT8U	Samples are synchronized by clock signals
SmpRate	INT16U	OPTIONAL; sample rate from the instance of MSVCB or USVCB
^a The type and value of this parameter shall be derived from the attribute OptFlds of the respective USVCB or MSVCB .		

2

3 **16.4.1 MsvID or UsvID**

4 The parameter **MsvID** or **UsvID** shall contain the values of the attributes **MsvID** or **UsvID** of
5 the **MSVCB** or **USVCB** to be included in the sampled value message.

6 **16.4.2 OptFlds**

7 The parameter **OptFlds** shall specify which of the optional fields (**RefrTm**, **SmpSynch** and
8 **SmpRate**) are included in the sampled value message. If the attribute of the sampled value
9 control block refresh-time (sample-rate, sample-synchronized) is TRUE then the field **RefrTm**
10 (**SmpSynch** or **SmpRate**) shall be contained in the sampled value message.

11 The parameter **OptFlds** shall be derived from the attribute **OptFlds** of the respective **USVCB**
12 or **MSVCB**.

13 **16.4.3 DatSet**

14 The parameter **DatSet** (taken from the **MsvID** or **UsvID**) shall contain the **ObjectReference**
15 of the **DATA-SET** whose values of the members are transmitted in the message.

16 **16.4.4 Sample [1..n]**

17 The parameter **Sample** shall contain the value of a member of **DATA-SET** sampled at
18 a given time.

19 **16.4.5 SmpCnt**

20 The parameter **SmpCnt** shall contain the values of a counter, which is incremented each time
21 a new sample of the analogue value is taken. The sample values shall be kept in the right or-
22 der. If the counter is used to indicate time consistency of various sampled values, the counter
23 shall be reset by an external synchronization event.

24 NOTE The external synchronization event is outside this part of the standard; details can be found in a SCSM.

1 **16.4.6 RefrTm**

2 The parameter **RefrTm** shall contain the time when the transmission buffer has been refreshed
3 locally.

4 NOTE The semantic of the **RefrTm** is defined in the SCSM. This time may be used by the subscriber to check the
5 validity of the data.

6 **16.4.7 ConfRev**

7 The parameter **ConfRev** shall contain the value of the attribute **ConfRev** of the **MSVCB** or
8 **USVCB**.

9 **16.4.8 SmpSynch**

10 The parameter **SmpSynch** shall indicate whether the sampled analogue values sent by the
11 **MSVCB** or **USVCB** are synchronized by clock signals. The following values shall be used:

12 2 = indicates that the SV are synchronized by a global area clock signal

13 1 = indicates that the SV are synchronized by a local area clock signal

14 0 = indicates that the SV are not synchronized by an external clock signal

15 **16.4.9 SmpRate**

16 The parameter **SmpRate** shall contain the value of the attribute **SmpRate** of the **MSVCB** or
17 **USVCB**.

18

1 17 CONTROL class model

2 17.1 Introduction

3 The control model provides a specific way to change the state of internal and external processes by a client. The control model can only be applied to **DATA** that has a `ctlModel` attribute. Such **DATA** will be referred to as a "**control object**".

6 The control model consists of

- 7 – specification of services;
- 8 – a behaviour described with state machines.

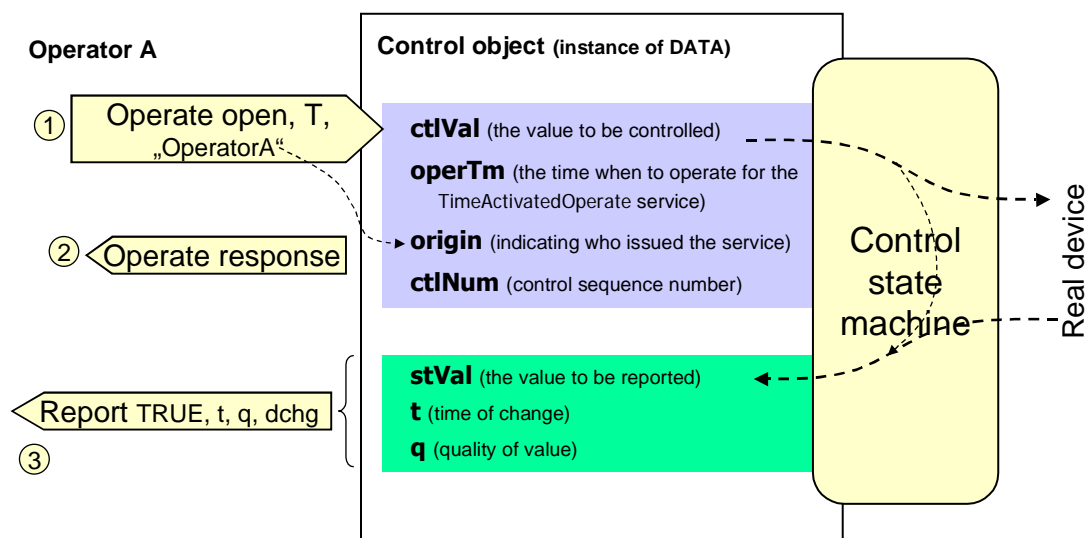
9 The control model defines the following services:

- 10 – **Select (Sel) / SelectWithValue (SelVal)**
- 11 – **Cancel**
- 12 – **Operate (Oper) / TimeActivatedOperate (TimOper)**
- 13 – **CommandTermination (CmdTerm)**

14

15 NOTE The abbreviations for these services may be used in the SCSM.

16 The concept of the control model is depicted in the example in Figure 35.



IEC 424/03

17

18

Figure 35 – Principle of the control model

19 The client (Operator A) issues the **Operate** service which is immediately confirmed by the **Operate** response. The new state change is reported by an independent **Report** indicating the final result of the control operation.

22 The services **Select**, **SelectWithValue**, **Cancel**, **Operate**, **TimeActivatedOperate**, and **CommandTermination** are related. The behaviour of these services shall be as defined in the state machines contained in this clause.

25 Depending on the application, different behaviours of a control object shall be used. Therefore, different state machines are defined. For a specific control object, the used model shall be defined in a configuration parameter. Four cases are defined:

27

- 1 Case 1: Direct control with normal security (**direct-operate**)
- 2 Case 2: SBO control with normal security (**operate-once** or **operate-many**)
- 3 Case 3: Direct control with enhanced security (**direct-operate**)
- 4 Case 4: SBO control with enhanced security (**operate-once** or **operate-many**)
- 5 As shown in the state diagrams, the change from one state to the next state
- 6 by the parameter "check condition". The check condition may be specified
- 7 meter (for example, synchrocheck). Besides the check condition specific
- 8 parameter, the control object may perform additional checks.

shall perform additional checks such as checking Logical Node Behavior and remote/local condition etc. The proper control response shall be generated according to the used model (e.g., direct control with normal security).

9 **17.2 Control with normal security**

10 In the case of control with normal security there shall be no additional supervision of the status

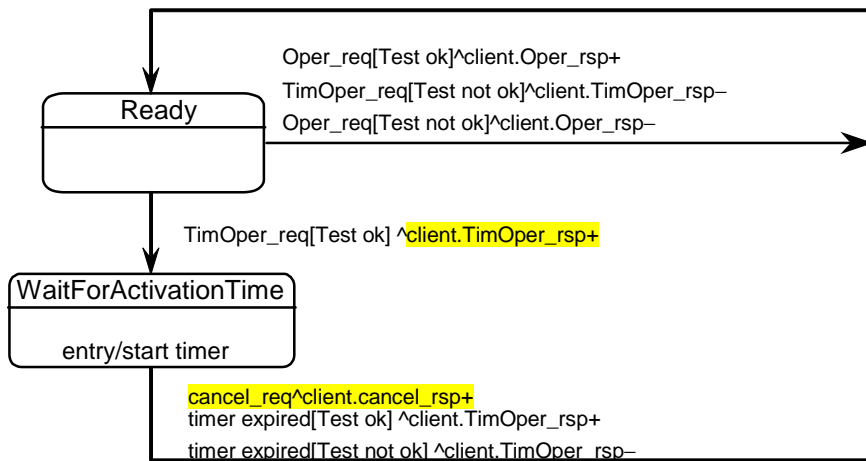
11 value by the control object. This means that for the negative case, if the status value did not

12 change to the control value, the client will not get information about the failure from the control

13 object.

14 **17.2.1 Direct control with normal security**

15 This model shall use the services **Operate** and **TimeActivatedOperate**.



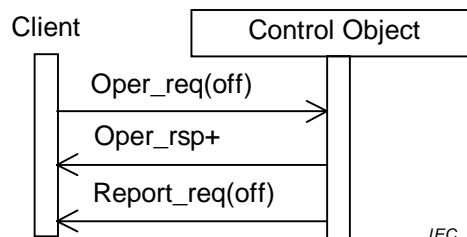
IEC 425/03

16
17
18 **Figure 36 – State machine of direct control with normal security**

19 Direct control with normal security should be used for operations that act either on local **DATA**

20 (for example, a LED test) or on **DATA** that influence external devices where a return informa-

21 tion is not supervised (for example, switch on a heating).



IEC 426/03

22
23 **Figure 37 – Direct control with normal security**

1 **Procedure**

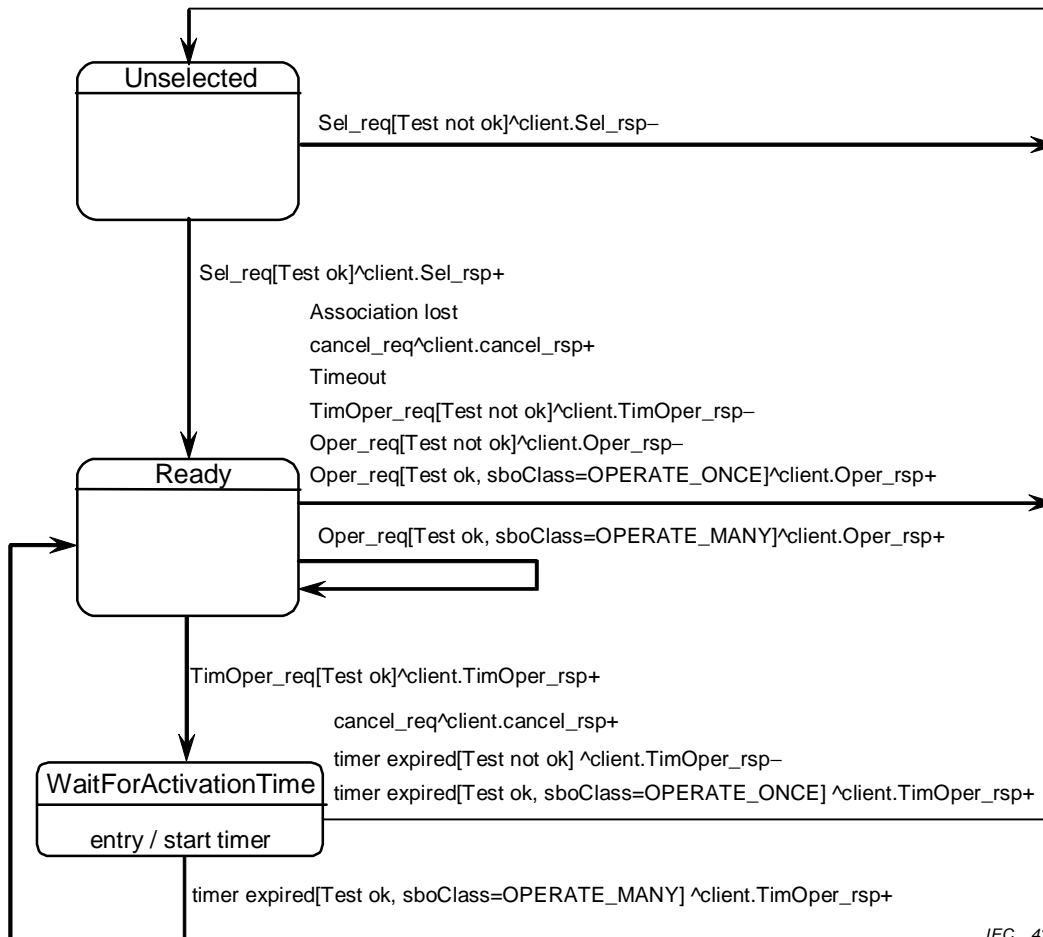
2 On receipt of an **Operate** request, the control object shall check validation of the control
3 execution.

- 4 • If not successful, the control object shall issue a negative response to the requesting cli-
5 ent.
6 • If successful, the control object shall issue a positive response to the requesting client and
7 causes the requested action.

8 The new status may be reported by the **Report** service (see reporting model).

9 **17.2.2 SBO control with normal security**

10 This model shall use the services **Select**, **Cancel**, **Operate**, and **TimeActivatedOper-**
11 **ate**. Once not in the unselected state, only the client that selected the control object shall be
12 able to force state transitions for the control object. All control object requests, for the selected
13 control object from other clients shall generate a negative response.



14

IEC 427/03

15 NOTE This state machine is compatible to the SBO control model defined in UCA™.2.

16 **Figure 38 – State machine of SBO control with normal security**

17

1 Procedure

- 2 a) On receipt of a **Select** request, the control object shall determine if the client has appropri-
3 ate access authority, that the control object is not currently selected by a different client,
4 and that the device represented by the associated **LOGICAL-NODE** is operable and is not
5 tagged so as to restrict operation.
- 6 – If the **Select** operation is not valid, the control object shall issue a negative response to
7 the requesting client.
- 8 – If the **Select** operation is valid, the control object shall issue a positive response to the
9 requesting client, shall change the state to ready and starts a deselect timer for either
10 the interval defined by the SelTimOut attribute or, if unimplemented, some locally de-
11 termined duration.
- 12 b) If the deselect timer expires before an **Operate** request on one or more of the other control
13 components shall be requested by the selecting client, the control object shall change the
14 state to unselected.
- 15 c) If an **Operate** request is received from the selecting client while the state is unselected for
16 that client, the operation shall be denied.
- 17 d) On receipt of an **Operate** request, the control object shall check validation of the control
18 execution.
- 19 – If not successful, the control object shall issue a negative response to the requesting
20 client.
- 21 – If successful, the control object shall issue a positive response to the requesting client
22 and shall cause the requested action by activating a binary output (or sending an
23 equivalent signal on a process bus).

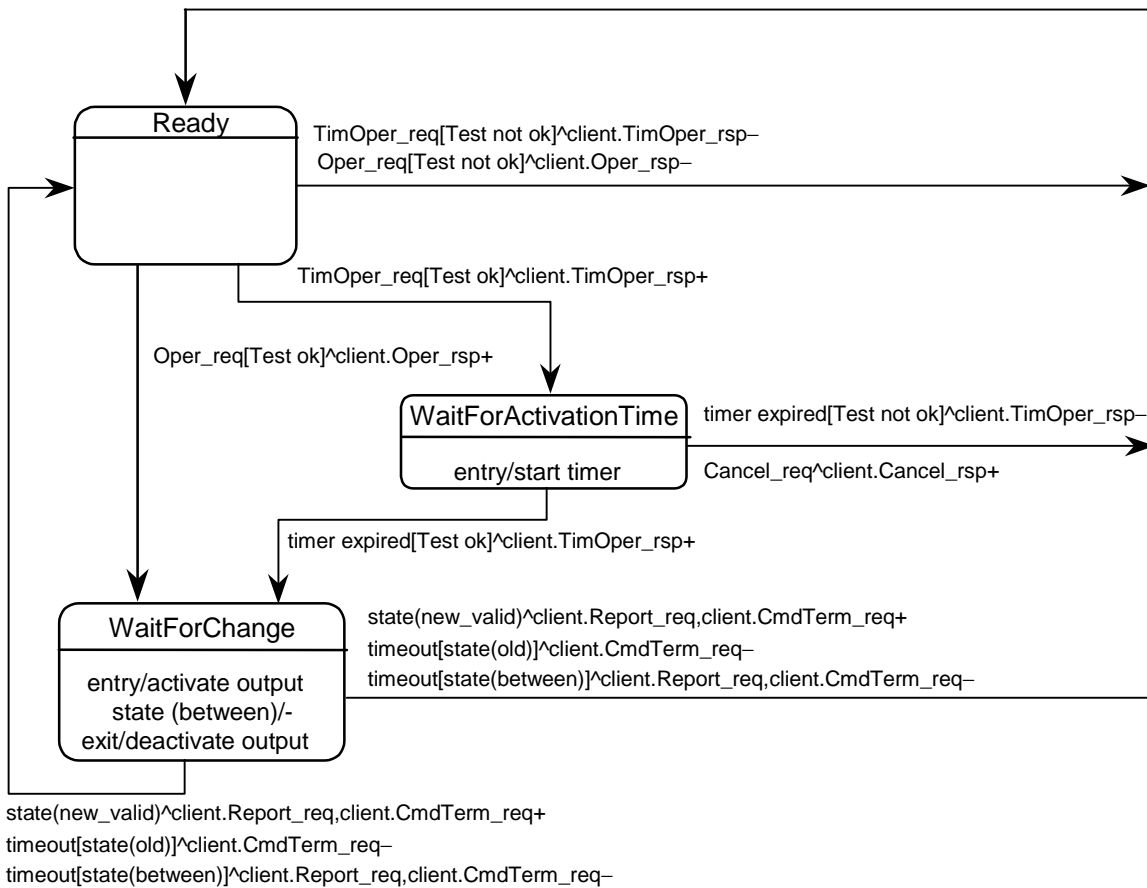
24 17.3 Control with enhanced security

25 17.3.1 Introduction

26 In the case of control with enhanced security there shall be an additional supervision of the
27 status value by the control object. Each command sequence shall be terminated by a **Com-**
28 **mandTermination** service primitive.

29 17.3.2 Direct control with enhanced security

30 This model shall use the services **Operate**, **TimeActivatedOperate**, and **Command-**
31 **Termination**.



1
2

IEC 428/03

3

Figure 39 – State machine of direct control with enhanced security

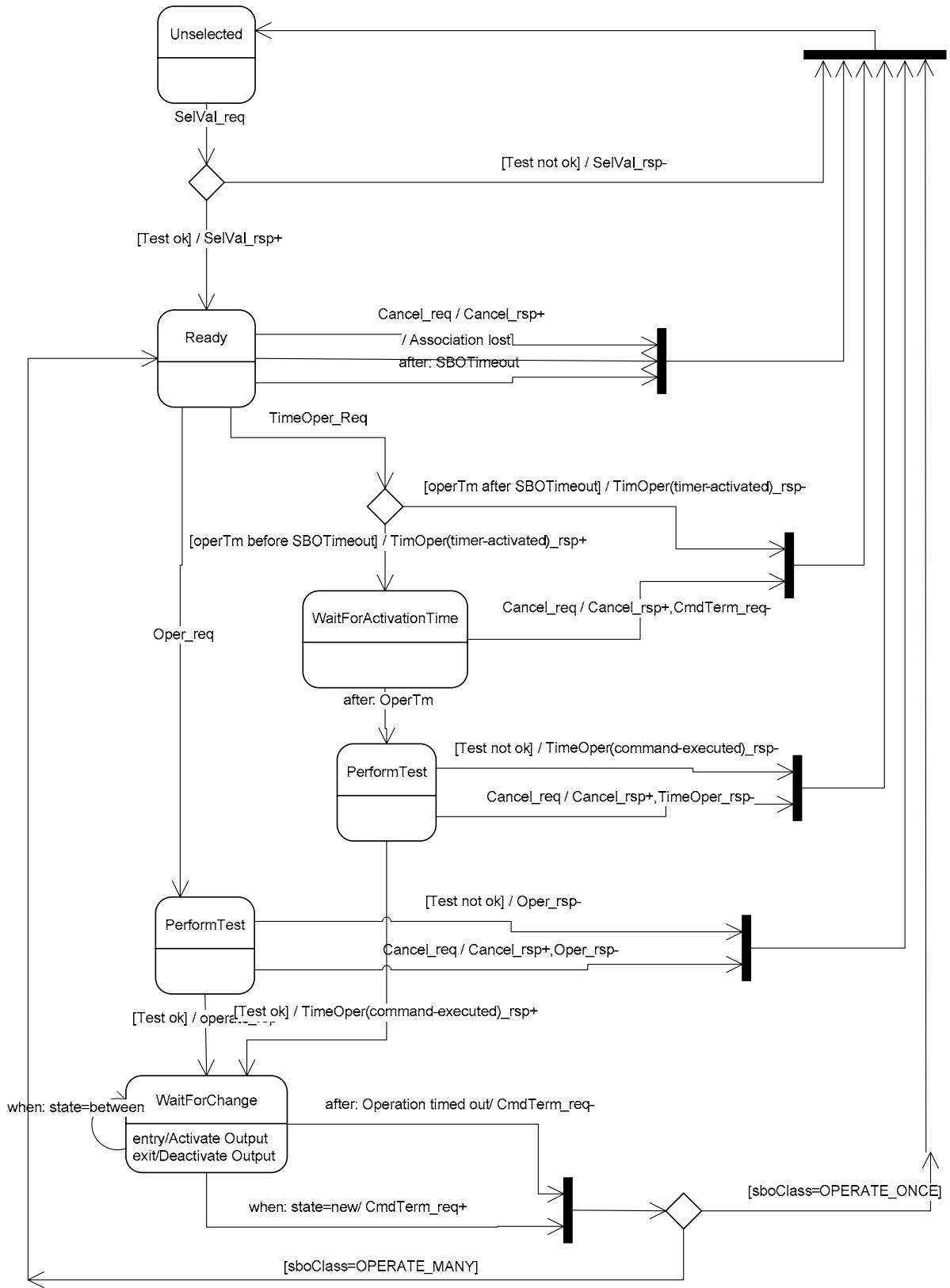
4
5

NOTE – as a result of the state change, the status attribute of the control object may change and this change may be e.g. reported. In the following sequence diagrams, such a report is shown.

6 **17.3.3 SBO control with enhanced security**

7
8
9
10

This model uses the services **SelectWithValue**, **Cancel**, **Operate**, **TimeActivatedOperate**, and **CommandTermination**. Once not in the unselected state, only the client that selected the control object shall be able to force state transitions for the control object. All control object requests, for the selected control object from other clients shall generate a negative response.

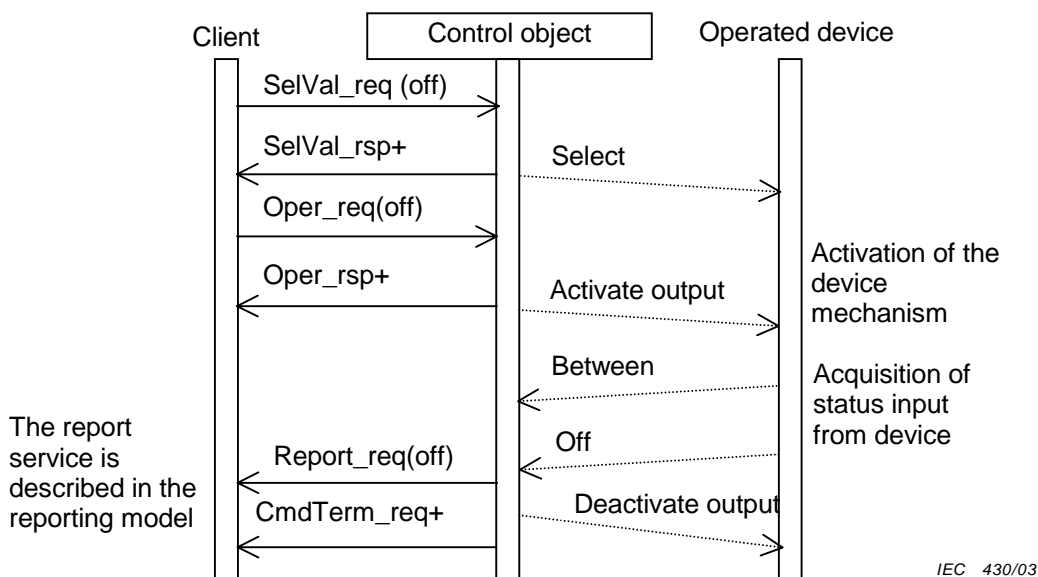


1
2
3
4
5

Figure 40 – State machine SBO control with enhanced security

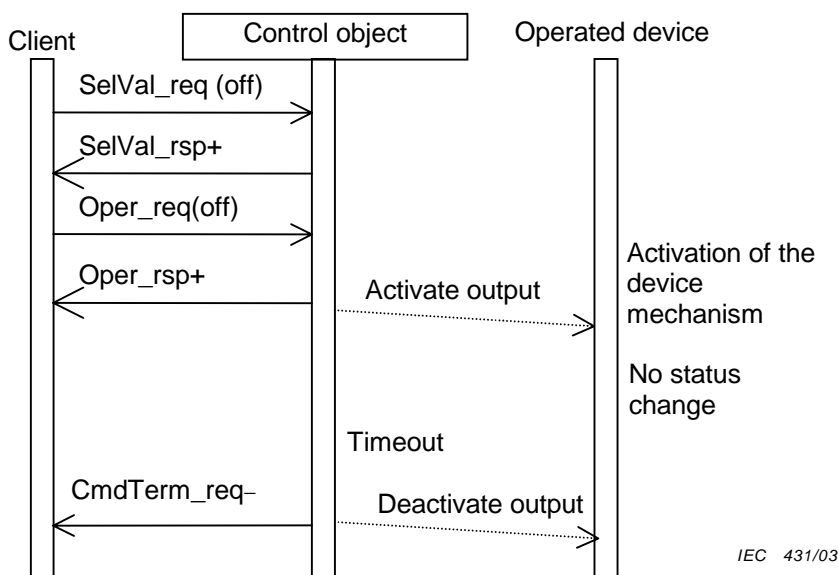
NOTE – as a result of the state change, the status attribute of the control object may change and this change may be e.g. reported. In the following sequence diagrams, such a report is shown.

- 1 Control with enhanced security should be used for control procedures that cause an important
- 2 action outside the device containing the accessed control object.



3
4 **Figure 41 – Select before operate with enhanced security – positive case**

- 5 NOTE The dashed lines in Figure 41 and Figure 42 indicate that these “services” are local and not visible at the
- 6 communication level.



7
8 **Figure 42 – Select before operate with enhanced security –**
9 **negative case (no status change)**

10 **Procedure**

- 11 a) On receipt of a **SelectWithValue** request, the control object shall determine if the client
- 12 has appropriate access authority, that the control object is not currently selected by a differ-
- 13 ent client, and that the device represented by the associated **LOGICAL-NODE** is operable
- 14 and is not tagged so as to restrict operation.
- 15 – If the **SelectWithValue** operation is not valid, the control object shall issue a negative
- 16 response to the requesting client.

- 1 – If the **SelectWithValue** operation is valid, the control object shall issue a positive re-
2 response to the requesting client, shall change the state to ready and starts a deselect
3 timer for either the interval defined by the **sboTimOut** attribute or, if unimplemented,
4 some locally determined duration.
- 5 b) If the deselect timer expires before an **Operate** request on one or more of the other control
6 components shall be requested by the selecting client, the control object shall change the
7 state to unselected.
- 8 c) If an **Operate** request is received from the selecting client while the state is not Ready for
9 that client, the operation shall be denied.
- 10 d) On receipt of an **Operate** request, the control object shall check validation of the control
11 execution.
- 12 – If not successful, the control object shall issue a negative response to the requesting
13 client.
- 14 – If successful, the control object shall issue a positive response to the requesting client
15 and shall cause the requested action by activating a binary output (or sending an
16 equivalent signal on a process bus). The control object shall turn to the state WaitFor-
17 Change.
- 18 – The control object supervises the change of the device status.
- 19 – As soon as the status of the controlled device has changed, the control object shall re-
20 port the new status using the report service of the reporting model.
- 21 – If the status has not changed to the wanted value after a certain time, the control object
22 shall issue a CommandTermination negative as soon as the output is deactivated.
- 23 – When the object indicates the wanted position before expiration of a timer, the control
24 object shall issues a CommandTermination positive as soon as the output is deacti-
25 vated.
- 26 e) When leaving the **WaitForChange** state, one of the following procedures shall be performed
27 based on the SBO-Select Class.
- 28 – If the value of the **sboClass** attribute is **operate-once**, the new state shall be unse-
29 lected.
- 30 – If the value of the **sboClass** attribute is **operate-many**, the new state shall be Ready.
- 31 The last action shall be the command termination (**CmdTerm**) service.

33 **17.4 Time-activated operate**

34 Time-activated control shall consist of a **TimeActivatedOperate** request and response. The
35 response shall inform the requesting client whether the command was successful, and had
36 caused a time activation process, or unsuccessful.

37 This shall be an extension of the control model. To use the time-activated operate capability
38 the service Operate in the control model shall be replaced by the service **TimeActivated-**
39 **Operate**.

40 NOTE The example below is shown with the sboClass direct-operate. The use of select before operate mode is
41 also possible. In that case, the control object must be in the state Ready before the service TimeActivatedOperate
42 is supported.

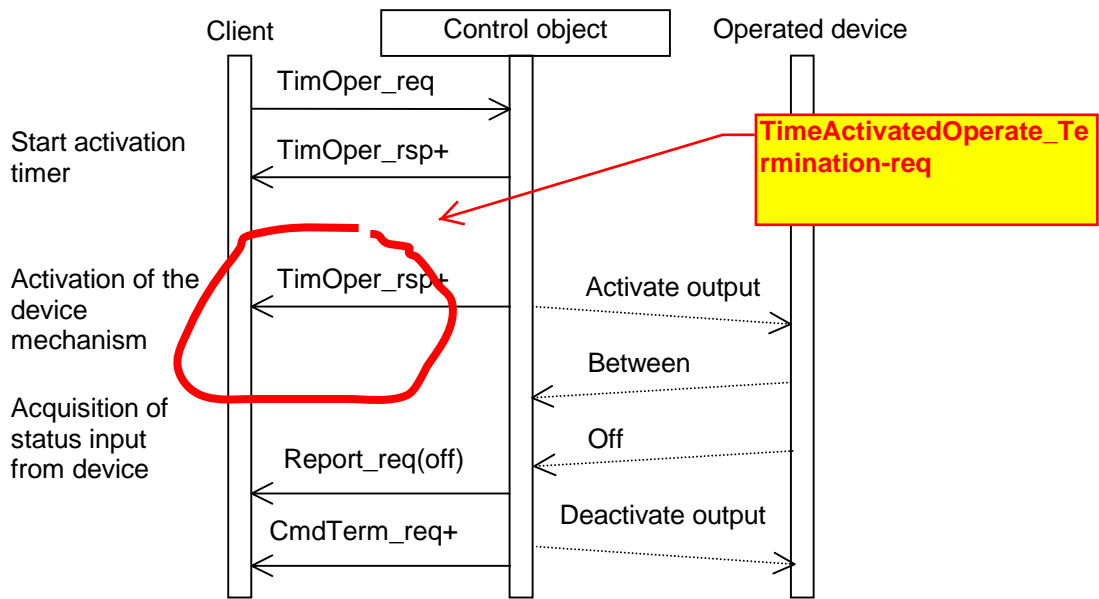


Figure 43 – Time-activated operate

IEC 432/03

1

2

3 Procedure

- 4 a) On receipt of a **TimeActivatedOperate** request the control object shall check the validity.
- 5 – If not successful, the control object shall send a negative response to the requesting cli-
- 6 ent.
- 7 – If successful, the control object shall activate the timer and shall send a positive re-
- 8 sponse with the information that the timer was started.
- 9 b) On expiration of the timer the wanted action shall be activated and a response shall be sent
- 10 to the client.
- 11 c) All further information exchange shall be as described in the model for control with en-
- 12 hanced security.

13 **17.5 CONTROL class service definitions**

14 **17.5.1 Overview**

15 For **CONTROL** the following services listed in Table 35 are defined.

16

Table 36 – Control services

ACSI control service
Select (Sel)
SelectWithValue (SelVal)
Cancel (Cancel)
Operate (Oper)
CommandTermination (CmdTerm)
TimeActivatedOperate (TimOper)

17

1 17.5.2 Service parameter definition

2 The following service parameters shall be applied in the service definitions. Additional service
3 parameters for control may be defined with the respective common data classes.

4 17.5.2.1 ControlObjectReference

5 The parameter **ControlObjectReference** shall contain the ObjectReference of the controllable
6 **DATA** (defined in IEC 61850-7-4) to be accessed, for example **Pos**, which represents the
7 **DATA** “**Position**”.

8 ~~17.5.2.2 AddCause – additional cause diagnosis~~

9 The parameter **AddCause** shall identify the reason for failure in a negative control service
10 specific response. The values shall be as defined in Table 37.

11 **Table 37 – Additional cause diagnosis definition**

Additional cause diagnosis type		
Attribute name	Attribute type	Value/value range/explanation
AddCause	ENUMERATED	ServiceError type Blocked-by-switching-hierarchy Select-failed Invalid-position Position-reached Parameter-change-in-execution Step-limit Blocked-by-Mode Blocked-by-process Blocked-by-interlocking Blocked-by-synchrocheck Command-already-in-execution Blocked-by-health nof-n-control Abortion-by-cancel Time-limit-over Abortion-by-trip Object-not-selected

12
13 The description of the values shall be as defined in Table 38.

14 **Table 38 – AddCause semantic**

Value	Explanation
ServiceError type	All errors as defined in Table 5
Blocked-by-switching-hierarchy	Not successful since one of the downstream Loc switches like in CSWI has the value TRUE
Select-failed	Cancelled due to an unsuccessful selection (select service)
Invalid-position	Control action is aborted due to invalid switch position (Pos in XCBR or XSWI)
Position-reached	Switch is already in the intended position (Pos in XCBR or XSWI)
Parameter-change-in-execution	Control action is blocked due to running parameter change
Step-limit	Control action is blocked, because tap changer has reached the limit (EndPosR or EndposL in YLTC)
Blocked-by-Mode	Control action is blocked, because the Mod (CSWI or XCBR/XSWI) is in a mode (Mod) which does not allow any switching
Blocked-by-process	Control action is blocked due to some external event at process level that prevents a successful operation, for example, blocking indication (FEHealth in XCBR or XSWI)
Blocked-by-interlocking	Control action is blocked due to interlocking of switching devices (in CILO attribute EnaOpn.stVal ="FALSE" or EnaCls.stVal ="FALSE"

Value	Explanation
Blocked-by-synchrocheck	Control action with synchrocheck is aborted due to the exceeding of the time limit and missing synchronism condition
Command-already-in-execution	Control service or cancel is rejected, because control action is already running
Blocked-by-health	Control action is blocked due to some internal event that prevents a successful operation (Health)
1-of-n-control	Control action is blocked, because another control action in a domain (for example, substation) is already running (in any XCBR or XSWI , the DPC.stSeld="TRUE").
Abortion-by-cancel	Control action is aborted due to cancel service
Time-limit-over	Control action is terminated due to exceed of some time limit
Abortion-by-trip	Control action is aborted due to a trip (PTRC with ACT.general="TRUE")
Object-not-selected	Operation can not be executed, since object is not selected

1

2 **17.5.2.3 TimOperRsp – TimeActivatedOperate response**

3 This parameter **TimOperRsp** shall specify the details of the response on the service **TimeAc-**
 4 **tivatedOperate** as shown in Table 39

5 **Table 39 – TimeActivatedOperate response definition**

dafür jetzt ein neuer Service für den Fall command-executed.

TimeActivatedOperate response type		
Attribute name	Attribute type	Value/value range/explanation
TimOperRsp	ENUMERATED	timer-activated command-executed

6

7 **17.5.3 Service specification**

8 **17.5.3.1 General**

9 With the common data classes, additional service parameters for control are defined. The
 10 control services use these service parameters to communicate the specific values of the
 11 service. All service parameters of a specific service need to be included in one service re-
 12 quest.

13 NOTE 1 The SCSM defines the subset of service parameters in the response service primitives. A communication
 14 stack that allows the client to assign a response to the relating request may not support all the service parameters
 15 that were also transmitted in the request.

16 NOTE 2 The additional cause diagnosis is a service parameter that is only transmitted in the response service
 17 primitives. The SCSM defines how this service parameter is included in the response PDU.

18 **17.5.3.2 Select (Sel)**

19 The **Select** service shall define the following service parameters.

Parameter name
Request
ControlObjectReference
Response+
ControlObjectReference
Response–
ControlObjectReference

20

1 NOTE The service parameters are defined in 17.5.2.

2

3 **17.5.3.3 SelectWithValue (SelVal)**

4 The **SelectWithValue** service shall define the following service parameters.

Parameter name
Request
ControlObjectReference
Additional service parameters from common data class
Response+
ControlObjectReference
Additional service parameters from common data class
Response-
ControlObjectReference
Additional service parameters from common data class
AddCause

5

6 NOTE The service parameters are defined in 17.5.2.

7

8 **17.5.3.4 Cancel**

9 The **Cancel** service shall be used for the de-selection.

Change text to: "The cancel service shall be used to abort a control operation. It is indicated in the state machines, in which states it is possible to cancel the control operation."

Parameter name
Request
ControlObjectReference
Additional service parameters from common data class
Response+
ControlObjectReference
Additional service parameters from common data class
Response-
ControlObjectReference
Additional service parameters from common data class
AddCause

10

11 NOTE The service parameters are defined in 17.5.2.

12

1 **17.5.3.5 Operate (Oper)**

2 The **Operate** service shall define the following service parameters.

Parameter name
Request
ControlObjectReference
Additional service parameters from common data class
Response+
ControlObjectReference
Additional service parameters from common data class
Response-
ControlObjectReference
Additional service parameters from common data class
AddCause

3

4 NOTE The service parameters are defined in 17.5.2.

5

6 **17.5.3.6 CommandTermination (CmdTerm)**

7 The **CommandTermination** service shall define the following service parameters.

Parameter name
Request+
ControlObjectReference
Additional service parameters from common data class
Request-
ControlObjectReference
Additional service parameters from common data class
AddCause

8

9 NOTE The service parameters are defined in 17.5.2.

10

1 17.5.3.7 TimeActivatedOperate (TimOper)

2 The **TimeActivatedOperate** service shall define the following service parameters.

Parameter name
Request
ControlObjectReference
Additional service parameters from common data class
Response+
ControlObjectReference
Additional service parameters from common data class
TimOperRep
Response-
ControlObjectReference
Additional service parameters from common data class
TimOperRep
AddCause

timer-activated

3

4 NOTE The service parameters are defined in 17.5.2.

5

17.5.3.8 TimeActivatedOperate_Termination

TimOperRsp WERT = command-executed

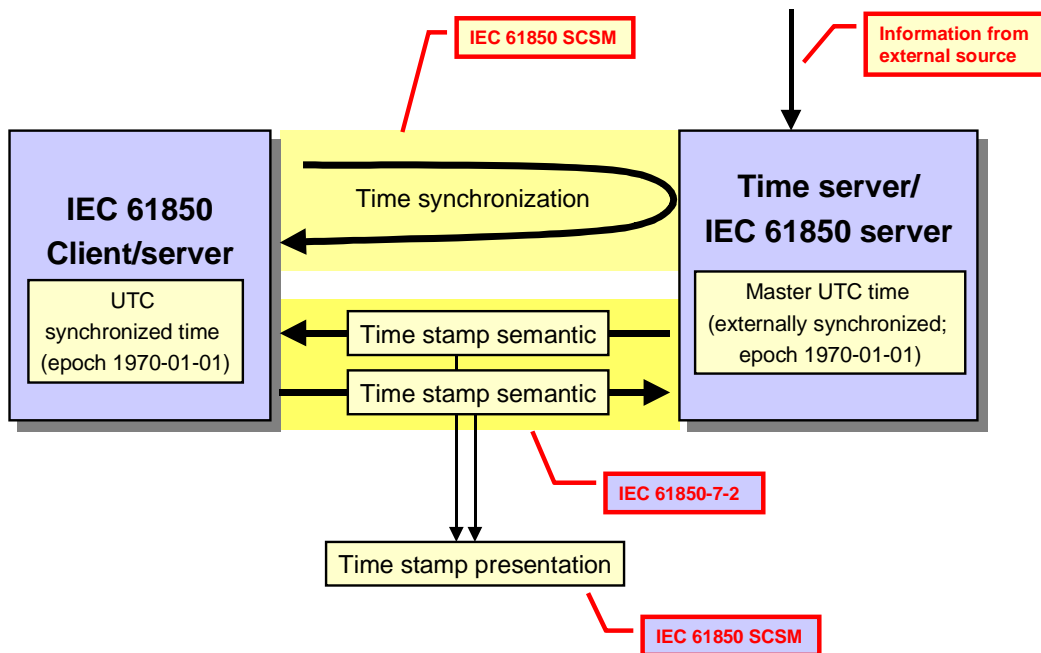
Das ist der Abschluss ...

Consequences in statemachines and sequence diagrams.

1 18 Time and time-synchronization model

2 18.1 General

3 The time and time-synchronization model shall provide the UTC synchronized time to applica-
 4 tions located in server and client substation IEDs. The components of the time and time-
 5 synchronization model are depicted in Figure 44.



6
 7 **Figure 44 – Time model and time synchronization (principle)** IEC 433/03

8 The model shall comprise

- 9 – the **external information** required by the **time master** from an external source to synchrono-
 10 nize other substation server or client IEDs (see 18.2);
- 11 – **time server** providing the source for the substation internal time synchronization and
 12 source for time stamping (in case the time server is implemented together with an
 13 IEC 61850 client/server in one physical device);
- 14 – **time synchronization** protocol providing time synchronization with other IEDs. Time syn-
 15 chronization shall meet the requirements of IEC 61850-5; the specification of time synchrono-
 16 zation is defined in the SCSMs (for example, SNTP for IEC 61850-8-1);
- 17 – the **time stamp semantics** used for information exchange of the ACSI (see 5.5.3.6);
- 18 – the **presentation** of the time stamps according to the chosen SCSM;
- 19 – the **server** and **clients** that need substation-wide synchronized time.

20 18.2 External information

21 External information required for the time and time synchronization model shall provide the fol-
 22 lowing.

23 a) Received external time

- 24 – synchronized time to some known level of accuracy;
- 25 – elapsed number of seconds since Epoch. If this count of seconds includes the leap
 26 seconds that have occurred since the epoch then the time produced by this time server
 27 shall have the LeapSecondsKnown quality attribute set to true, otherwise set to false.

28 b) **Epoch** (for example, GPS 6.1.1980).

1 19 Naming conventions

2 19.1 Class naming and class specializations

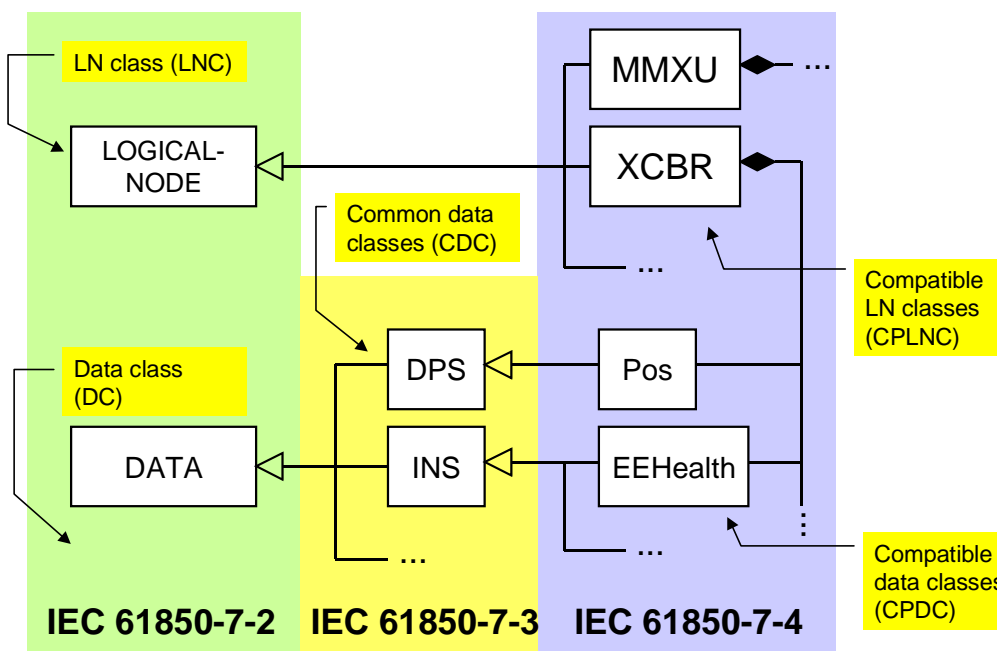
3 The classes for **DATA**, **common DATA**, **compatible DATA**, and **compatible LOGICAL-**
 4 **NODE** defined in IEC 61850-7-x make use of the following specializations:

5 IEC 61850-7-3 common **DATA** classes (for example, **DPC**) are specializations of the class
 6 **DATA** of IEC 61850-7-2

7 IEC 61850-7-4 compatible **DATA** classes (for example, **Pos** – position) are specializations of
 8 IEC 61850-7-3 common **DATA** classes (for example, **DPC** – controllable
 9 double point)

10 IEC 61850-7-4 compatible **LOGICAL-NODE** classes (for example, XCBR) are specializations
 11 of the **LOGICAL-NODE** class of IEC 61850-7-2

12 Figure 45 shows an overview of the specializations.



IEC 434/03

13
 14 **Figure 45 – Specializations**

15 Each class in IEC 61850-7-x has its own class name. These class names shall be the basic
 16 building blocks when referencing class instances.

1 **19.2 Referencing an instance of a class**

2 The **ObjectReferences** and the abbreviations (used in class and service definitions) shall be
 3 as listed in Table 40.

4 **Table 40 – List of ObjectReferences**

ACSI class	ObjectReference of instance
LOGICAL-DEVICE	
LDRef (logical device reference)	LDName
LOGICAL-NODE	
LNRef (logical node reference)	LDName/LNName
DATA	
DataRef (data reference)	LDName/LNName.DataName[. DataName[. ...]]
DataAttribute	
DATRef	LDName/LNName. DataName[. DataName[. ...]]. DataAttributeName[. DAComponentName[
DataAttributeReference (data attribute reference)	LDName/LNName. DataName[.DataName[. ...]]. DataAttributeName[NumArrayElement]
DATA-SET	
DSRef (data set reference)	LDName/LNName. DataSetName (persistent), or @DataSetName (non-persistent)
SETTING-GROUP-CONTROL	
SGCB-Reference	LDName/LLN0.SGCB
BUFFERED-REPORT-CONTROL-BLOCK	
BRCBRef (buffered report control block reference)	LDName/LNName.BRCBName
UNBUFFERED-REPORT-CONTROL-BLOCK	
URCBRef (unbuffered report control block reference)	LDName/LNName.URCBName
LOG-CONTROL	
LCBRef (log control block reference)	LDName/LNName.LCBName
LOG	
LogRef (log reference)	LDName/LNName.LogName
GOOSE	
GoCBRef (GOOSE control block reference)	LDName/LLN0.GoCBName
GSSE	
GsCBRef (GOOSE control block reference)	LDName/LLN0.GsCBName
MSVCB	
MsvCBRef (multicast sampled value control block)	LDName/LLN0.MsvCBNam
USVCB	
UsvCBRef (multicast sampled value control block)	LDName/LLN0.UsvCBNam

[(NumArrayElement)]
 "[]" for optional, "()" for
 number), if we have
 1000 elements, we
 need 6 Characters for
 the array element ,...
 do we need to
 constrain the number?

A similar table needs to be defined for the other names/references like log, lcb, ... these names are also not unlimited!! Zug, 2007-07-10

1 Additionally, the following length definitions shall apply.

LDName/LNName.
AttributeName[. ...].DataAttributeName[.DAComponentName[. ...]]

2 The inner square bracket “[. ...]” shall indicate further recursive defini-
3 tribute components.

(may start with any numeric character) - see MMS Edition 2003: page XIII: f) The restrictions on the characters that can be used as an Identifier have been relaxed to allow an Identifier to begin with a numeric character, and by extension, to consist solely of numeric characters.

4 **LDName** = up to 64 characters, application specific

5 **LNName** = [LN-Prefix] LN class name [LN-Instance-ID]

6 LN-Prefix = m characters (application specific)

7 LN class name = 4 characters (for example, compatible logical node name as defined in IEC 61850-7-4)

8 LN-Instance-ID = n numeric characters (application specific)

9 m+n ≤ 7 characters

The number can be any number. There is no rule, e.g., only consecutive numbers ... no data class should end with a numeric character ...

11 **DataName** = DataClassName[Data-Instance-ID]

12 **DataClassName** = up to 10 characters (as, for example, used in IEC 61850-7-4)

13 **Data-Instance-ID** = n numeric characters, optional; n shall be equal for all instances
14 of the same data class

15 **FCD** ≤ 61 characters including all separators “.” (without the value of the **FC**)

16 The characters allowed shall be:

Add FCDA (what about FCCB -> need to differentiate FCDA and FCCB here as well)

17 VisibleString (FROM

18 ("A" | "a" | "B" | "b" | "C" | "c" | "D" | "d" | "E" | "e" | "F" | "f" |
19 "G" | "g" | "H" | "h" | "I" | "i" | "J" | "j" | "K" | "k" | "L" | "l" |
20 "M" | "m" | "N" | "n" | "O" | "o" | "P" | "p" | "Q" | "q" | "R" | "r" |
21 "S" | "s" | "T" | "t" | "U" | "u" | "V" | "v" | "W" | "w" | "X" | "x" |
22 "Y" | "y" | "Z" | "z" | "_" | "0" | "1" | "2" | "3" | "4" | "5" | "6" |
23 "7" | "8" | "9"))

minus the (array number), element number is not known, because max number is a configurable value. Need to not here that 61 minus "(" "n" ")" reduces the usable range!

25 EXAMPLE Figure 46 shows examples of object names and object references. The example at the top
26 (first five lines) can be just five class definitions (not yet instantiated) or five instances of the classes
27 E1.QA5/XCBR.Pos.ctIVal, ...stVal, ...q, ...t, ...ctIModel. The object references in this case do not indicate if ob-
28 ject references refer to classes or instances. The context in which these references are used has to provide suffi-
29 cient information to know what is meant (just class or instance).

30 The other examples refer to instances only.

31 NOTE The LD name E1.QA5 and its structure are outside the scope of IEC 61850. The functional constraint (**FC**)
32 is not shown in the object reference. The FC information may be mapped into the **ObjectReference** in an SCSM;
33 IEC 61850-8-1 maps the FC between LN and Data.

ObjectReference type is VISIBLE STRING129, we need this extended set (compared to the VisibleString (FROM ("A" | "a" | "B" |) to define the "." and "/" as separators and "(" and ")" for describing an array element.

LD	LN	Data	DAttr.	FC	
E1.QA5	/XCBR	.Pos	.ctlVal	CO	Class or instance
E1.QA5	/XCBR	.Pos	.stVal	ST	
E1.QA5	/XCBR	.Pos	.q	ST	
E1.QA5	/XCBR	.Pos	.t	ST	
E1.QA5	/XCBR	.Pos	.ctlModel	CF	
LD5	/YPTR2	.Temp	.mVal.i .mVal.f	MX MX	Instance # 2
E1.QA5	/XCBR8	.Pos	.ctlVal	CO	Instance # 8
E1.QA5	/XCBR8	.Pos	.stVal	ST	
E1.QA5	/XCBR8	.Pos	.q	ST	
E1.QA5	/XCBR8	.Pos	.t	ST	
E1.QA5	/XCBR8	.Pos	.ctlModel	CF	
Object name		Object name			
Object name		Object name			
Object name		Object name			
Object reference					

IEC 435/03

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

Figure 46 – Object names and object reference

19.3 Scope

Server specific scope (instances are defined outside of all LDs but in the server) shall be defined using the “/” and up to 64 characters to the right.

EXAMPLE /ABC.xyz

Logical device specific scope (instances are defined inside a specific LD) shall be defined as up to 64 characters, then “/” followed by up to 64 characters to the right.

EXAMPLE Atlanta_110/XCBR.Pos

TPAA specific scope (instances are defined inside a specific TPAA) shall be defined using “@”, then “/” followed by up to 64 characters to the right.

EXAMPLE @/DataSet5 (for non-persistent DATA-SETs).

NOTE 1 The SCsMs may map the Reference to a flat numerical index or to a character string that is derived from the definition above. These character strings may comprise additional elements such as the functional constraint (FC).

NOTE 2 IEC 61850-6 gives additional definitions on how the application-specific character strings for logical devices can be built.

1 **20 File transfer**

2 **20.1 File transfer model**

3 The ACSI file transfer services shall provide the functionality for transferring files from and to
4 file stores and for managing file stores.

5 NOTE The ACSI file services and the structure of the ACSI file store are intentionally limited in scope to simplify
6 implementation in functional restricted devices. The ACSI file store addresses a single file format – sequential un-
7 structured binary – which may contain programs, data, or both. Any interpretation of the contents is by mutual
8 agreement of the systems involved.

9 The **FILE** shall have the structure as defined in Table 41.

10 **Table 41 – FILE class definition**

FILE class		
Attribute name	Attribute type	Value/value range/explanation
FileName	VISIBLE STRING255	
FileSize [0..1]	INT32U	
LastModified	TimeStamp	
Services GetFile SetFile DeleteFile GetFileAttributeValues		

11

12 **20.1.1 FileName**

13 The attribute **FileName** shall be the name of the file in the ACSI file store.

14 NOTE File names may be structured to differentiate file types, for example, disturbance records, programs, and
15 parameter and configuration data.

16 **20.1.2 FileSize [0..1]**

17 The attribute **FileSize** (in octets) shall be the length of a file in the file store.

18 NOTE In case the FileSize cannot be determined (for example, in the case of an on-the-fly created COMTRADE
19 file) the meaning and interpretation of the FileSize is outside the scope of this standard.

20 **20.1.3 LastModified**

21 The attribute **LastModified** shall be the time when the file was last modified.

1 **20.2 File services**2 **20.2.1 GetFile**3 **20.2.1.1 GetFile parameter**

4 The **GetFile** service shall be used by a client to transfer the contents of a file from the server
5 to the client.

Parameter name
Request
FileName
Response+
File-Data
Response–
ServiceError

6

7 **20.2.1.2 Request**8 **FileName**

9 The parameter **FileName** shall specify the name of the file being transferred.

10 **20.2.1.3 Response+**

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

13 **File-Data**

14 The parameter **File-Data** shall contain the data transferred; the type of file-data is OCTET
15 STRING.

16 **20.2.1.4 Response–**

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

19 **20.2.2 SetFile**20 **20.2.2.1 SetFile parameter**

21 The SetFile service shall be used by a client to transfer the contents of a file from the client to
22 the server.

Parameter name
Request
FileName
File-Data
Response+
Response–
ServiceError

23

1 **20.2.2.2 Request**2 **20.2.2.2.1 FileName**

3 The parameter **FileName** shall specify the name of the file being transferred.

4 **20.2.2.2.2 File-Data**

5 The parameter **File-Data** shall contain the data transferred; the type of file-data is OCTET
6 STRING.

7 **20.2.2.3 Response+**

8 The parameter **Response+** shall indicate that the service request succeeded. The type for this
9 parameter is SCSM specific.

10 **20.2.2.4 Response–**

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

13 **20.2.3 DeleteFile**14 **20.2.3.1 DeleteFile parameter**

15 The **FileDelete** service shall be used by a client to delete a file in the file store of a server.

Parameter name
Request
FileName
Response+
Response–
ServiceError

16

17 **20.2.3.2 Request**18 **FileName**

19 The parameter **FileName** shall specify the name of the file being deleted.

20 **20.2.3.3 Response+**

21 The parameter **Response+** shall indicate that the service request succeeded. The type for this
22 parameter is SCSM specific.

23 **20.2.3.4 Response–**

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

1 20.2.4 GetFileAttributeValues

2 20.2.4.1 GetFileAttributeValues parameter

3 The **GetFileAttributeValues** service shall be used by a client to obtain the name and attrib-
4 utes of a specific file in the server's file store.

Parameter name
Request
FileName
Response+
FileName
FileSize [0..n]
LastModified
Response–
ServiceError

5

6 20.2.4.2 Request

7 **FileName**

8 The parameter **FileName** shall, when present, specify the name of the file whose attributes are
9 requested to be returned to the client.

10 20.2.4.3 Response+

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

13 20.2.4.3.1 **FileName**

14 The parameter **FileName** shall provide the name of the file whose attributes are returned.

15 20.2.4.3.2 **FileSize [0..n]**

16 The parameter **FileSize** shall contain attribute information **FileSize** describing the selected
17 file. This information consists of the size of the file..

18 20.2.4.3.3 **LastModified**

19 The parameter **FileAttribute** shall contain attribute information **LastModified** describing
20 the selected file. This information consists of the time of the last modification.

21 20.2.4.4 Response–

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

24

Annex A (normative)

ACSI conformance statement

A.1 General

The following ACSI conformance statements shall be used to provide an overview and details about a device claiming conformance with ACSI:

- ACSI basic conformance statement
- ACSI models conformance statement
- ACSI service conformance statement

to specify the communication features mapped to an SCSM.

NOTE 1 The conformance statements of this annex are abstract in the sense that the ACSI models and their services are mapped to application layer models, services, and protocols. Additional details on the conformance are defined in the SCSM.

NOTE 2 For several features the conformance requirement is implicitly defined with the common data class contained in IEC 61850-7-3 and the compatible **LOGICAL-NODE** classes and **DATA** classes contained in IEC 61850-7-4, for example, a TrgOp (trigger option) of the value qchg (quality change) of **DataAttribute** requires the support of the TrgOp (trigger option) qchg of the **BRCB** or **URCB**.

A.2 ACSI basic conformance statement

The basic conformance statement shall be as defined in Table A.1.

Table A.1 – Basic conformance statement

		Client/ subscriber	Server/ publisher	Value/ comments
Client-server roles				
B11	Server side (of TWO-PARTY-APPLICATION-ASSOCIATION)	–	c1	
B12	Client side of (TWO-PARTY-APPLICATION-ASSOCIATION)	c1	–	
SCSMs supported				
B21	SCSM: IEC 61850-8-1 used			
B22	SCSM: IEC 61850-9-1 used			
B23	SCSM: IEC 61850-9-2 used			
B24	SCSM: other			
Generic substation event model (GSE)				
B31	Publisher side	–	O	
B32	Subscriber side	O	–	
Transmission of sampled value model (SVC)				
B41	Publisher side	–	O	
B42	Subscriber side	O	–	

	Client/ subscriber	Server/ publisher	Value/ comments
c1 – shall be 'M' if support for LOGICAL-DEVICE model has been declared.			
O – Optional			
M – Mandatory			

1 A.3 ACSI models conformance statement

2 The ACSI models conformance statement shall be as defined in Table A.2.

3 Table A.2 – ACSI models conformance statement

		Client/ subscriber	Server/ publisher	Value/ comments
If Server side (B11) supported				
M1	Logical device	c2	c2	
M2	Logical node	c3	c3	
M3	Data	c4	c4	
M4	Data set	c5	c5	
M5	Substitution	O	O	
M6	Setting group control	O	O	
	Reporting			
M7	Buffered report control	O	O	
M7-1	sequence-number			
M7-2	report-time-stamp			
M7-3	reason-for-inclusion			
M7-4	data-set-name			
M7-5	data-reference			
M7-6	buffer-overflow			
M7-7	entryID			
M7-8	BufTm			
M7-9	IntgPd			
M7-10	GI			
M7-11	conf-revision			
M8	Unbuffered report control	O	O	
M8-1	sequence-number			
M8-2	report-time-stamp			
M8-3	reason-for-inclusion			
M8-4	data-set-name			
M8-5	data-reference			
M8-6	BufTm			
M8-7	IntgPd			
M8-8	GI			
M8-9	conf-revision			
	Logging	O	O	
M9	Log control	O	O	
M9-1	IntgPd			
M10	Log	O	O	
M11	Control	M	M	

		Client/ subscriber	Server/ publisher	Value/ comments
If GSE (B31/B32) is supported				
	GOOSE	O	O	
M13	GSSE	O	O	
If SVC (B41/B42) is supported				
M14	Multicast SVC	O	O	
M15	Unicast SVC	O	O	
For all IEDs				
M16	Time	M	M	Time source with required accuracy shall be available
M17	File Transfer	O	O	
c2 – shall be 'M' if support for LOGICAL-NODE model has been declared. c3 – shall be 'M' if support for DATA model has been declared. c4 – shall be 'M' if support for DATA-SET , Substitution, Report, Log Control, or Time model has been declared. c5 – shall be 'M' if support for Report, GSE, or SV models has been declared. M – Mandatory				

1

2 A.4 ACSI service conformance statement

3 The ACSI service conformance statement shall be as defined in Table A.3 (depending on the
4 statements in Table A.1).

5 **Table A.3 – ACSI service conformance statement**

	Services	AA: TP/MC	Client/ subscriber	Server/ publisher	Comments
Server (Clause 6)					
S1	ServerDirectory	TP		M	
Application association (Clause 7)					
S2	Associate		M	M	
S3	Abort		M	M	
S4	Release		M	M	
Logical device (Clause 8)					
S5	LogicalDeviceDirectory	TP	M	M	
Logical node (Clause 9)					
S6	LogicalNodeDirectory	TP	M	M	
S7	GetAllDataValues	TP	O	M	
Data (Clause 10)					
S8	GetDataValue	TP	M	M	
S9	SetDataValue	TP	O	O	
S10	GetDataDirectory	TP	O	M	
S11	GetDataDefinition	TP	O	M	

1

Table A.3 (continued)

	Services	AA: TP/MC	Client/ subscriber	Server/ publisher	Comments
Data set (Clause 11)					
S12	GetDataSetValues	TP	O	M	
S13	SetDataSetValues	TP	O	O	
S14	CreateDataSet	TP	O	O	
S15	DeleteDataSet	TP	O	O	
S16	GetDataSetDirectory	TP	O	O	

Substitution (Clause 12)					
S17	SetDataValue	TP	M	M	

Setting group control (Clause 13)					
S18	SelectActiveSG	TP	O	O	
S19	SelectEditSG	TP	O	O	
S20	SetSGValues	TP	O	O	
S21	ConfirmEditSGValues	TP	O	O	
S22	GetSGValues	TP	O	O	
S23	GetSGCBValues	TP	O	O	

Reporting (Clause 14)					
Buffered report control block (BRCB)					
S24	Report	TP	c6	c6	
S24-1	data-change (dchg)				
S24-2	qchg-change (qchg)				
S24-3	data-update (dupd)				
S25	GetBRCBValues	TP	c6	c6	
S26	SetBRCBValues	TP	c6	c6	
Unbuffered report control block (URCB)					
S27	Report	TP	c6	c6	
S27-1	data-change (dchg)				
S27-2	qchg-change (qchg)				
S27-3	data-update (dupd)				
S28	GetURCBValues	TP	c6	c6	
S29	SetURCBValues	TP	c6	c6	
c6 – shall declare support for at least one (BRCB or URCB).					

Logging (Clause 14)					
Log control block					
S30	GetLCBValues	TP	M	M	
S31	SetLCBValues	TP	O	M	
Log					
S32	QueryLogByTime	TP	c7	M	
S33	QueryLogAfter	TP	c7	M	
S34	GetLogStatusValues	TP	M	M	
c7 – shall declare support for at least one (QueryLogByTime or QueryLogAfter).					

1

Table A.3 (continued)

	Services	AA: TP/MC	Client/ subscriber	Server/ publisher	Comments
Generic substation event model (GSE) (14.3.5.3.4)					
GOOSE-CONTROL-BLOCK					
S35	SendGOOSEMessage	MC	c8	c8	
S36	GetGoReference	TP	O	c9	
S37	GetGOOSEElementNumber	TP	O	c9	
S38	GetGoCBValues	TP	O	O	
S39	SetGoCBValues	TP	O	O	
GSSE-CONTROL-BLOCK					
S40	SendGSSEMessage	MC	c8	c8	
S41	GetGsReference	TP	O	c9	
S42	GetGSSEElementNumber	TP	O	c9	
S43	GetGsCBValues	TP	O	O	
S44	SetGsCBValues	TP	O	O	
c8 – shall declare support for at least one (SendGOOSEMessage or SendGSSEMessage). c9 – shall declare support if TP association is available.					

Transmission of sampled value model (SVC) (Clause 16)					
Multicast SVC					
S45	SendMSVMessage	MC	c10	c10	
S46	GetMSVCBValues	TP	O	O	
S47	SetMSVCBValues	TP	O	O	
Unicast SVC					
S48	SendUSVMessage	TP	c10	c10	
S49	GetUSVCBValues	TP	O	O	
S50	SetUSVCBValues	TP	O	O	
c10 – shall declare support for at least one (SendMSVMessage or SendUSVMessage).					

Control (17.5.1)					
S51	Select		M	O	
S52	SelectWithValue	TP	M	O	
S53	Cancel	TP	O	O	
S54	Operate	TP	M	M	
S55	Command-Termination	TP	M	O	
S56	TimeActivated-Operate	TP	O	O	

File transfer (Clause 20)					
S57	GetFile	TP	O	M	
S58	SetFile	TP	O	O	
S59	DeleteFile	TP	O	O	
S60	GetFileAttributeValues	TP	O	M	

2

1

Table A.3 (continued)

	Services	AA: TP/MC	Client/ subscriber	Server/ publisher	Comments
Time (5.5)					
T1	Time resolution of internal clock				Nearest negative power of 2 in seconds
T2	Time accuracy of internal clock				T0
					T1
					T2
					T3
					T4
					T5
T3	Supported TimeStamp resolution				Nearest value of 2^{*-n} in seconds according to 5.5.3.7.3.3

2

1 **Bibliography**

2 IEEE-SA TR 1550-1999 – Utility Communications Architecture (UCA™) Version 2 ².

3 UTC (Universal Time Coordinated):

4 http://en.wikipedia.org/wiki/Coordinated_Universal_Time

² UCA™ is a registered trade mark of EPRI, Palo Alto (USA).

1

Index

2	Access control	30	43	GSSE service definitions	131
3	access restriction	31	44	Integrity	97
4	access view	30	45	integrity period.....	86
5	active buffer	69	46	LeapSeconds.....	25
6	additional cause diagnosis	161	47	log model	107
7	Application association model	30	48	Logical node class	39
8	ARRAY type	22	49	Naming conventions	167
9	AuthenticationParameter	33, 36	50	ObjectName	21
10	Basic types	20	51	ObjectReference.....	21
11	best efforts.....	79	52	Operate	164
12	BRC.....	80	53	Packed list type	23
13	buffer time	84	54	persistent instances of DATA-SET.....	59
14	Buffering events.....	98	55	PHYCOMADDR type.....	22
15	Cancel	163	56	polling data.....	77
16	change-of-state notification	77	57	publisher	119
17	CommandTermination	164	58	purge buffer.....	88
18	Common ACSI types	21	59	quality-change	86
19	COMMON-DATA class	52	60	reason for inclusion	115
20	CompositeCDC.....	45	61	Referencing instances	168
21	CONTROL class	152	62	Report	89, 105
22	cyclic-integrity	86	63	report generation	95
23	DAComponentName.....	48	64	report identifier	83
24	Data class attributes	46	65	ReportFormat	90
25	Data class model	44	66	Sampled value format.....	149
26	Data set class model.....	59	67	Select.....	162
27	DataAttributeName.....	48	68	SelectWithValue	163
28	DataAttributeReference	48	69	sequence-of-events	77, 107
29	data-change.....	86	70	SERVER class.....	26
30	edit buffer	68	71	ServiceError type.....	22
31	EntryID type	22	72	setting group	68
32	EntryTime type.....	25	73	SimpleCDC	45
33	File transfer	171	74	subscribers.....	119
34	functional constraint.....	49	75	Substitution model	66
35	General interrogation	97	76	Time activated control.....	159
36	general-interrogation	87	77	Time stamp type	23
37	Generic substation event model	120	78	TimeActivatedOperate	165
38	GOOSE	119	79	TimeStamp type	23
39	GOOSE message.....	127	80	TrgOp and Reporting	51
40	GOOSE service Definitions	122	81	trigger option	50
41	GOOSE-CONTROL-BLOCK	120	82	trigger options enabled	86
42	GSSE.....	119, 129	83	TriggerConditions.....	50

1	TriggerConditions type	25	5	UTC	23
2	Type definitions.....	18	6	view.....	30
3	UNBUFFERED-REPORT-CONTROL-				
4	BLOCK	104			
7					
8					
9					
10					
11					
12					
13					