

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 CDV Ed2

2008-02-20

Dear All,

This is the first draft CDV to be discussed in Fredericia (Denmark), 25.-26.02.2008

This draft takes into account the Tissues on 7-2 (a few are still open; see email on the tissues for 7-2 dated 17.02.)

The discussions and decisions of the editors' group meeting in Zug have been taken into account.

A few official comments <<6185072ed2r100cdcc.doc>> still needs to be solved.

Comments included for the CD version are still contained in this version.

This document is distributed as pdf in two forms: one with comments visible and one without.

I look forward to seeing you in Denmark soon.

Regards,
Karlheinz Schwarz

Version: 61850-7-2_R1-01 Draft CDV Ed2_2008-02-20

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

- FOREWORD8
- INTRODUCTION 10
- 1 Scope11
- 2 Normative references 11
- 3 Terms and definitions 12
- 4 Abbreviated terms 13
- 5 ACSI overview and basic concepts 15
 - 5.1 General15
 - 5.2 Overview of basic information models15
 - 5.3 Overview of the other service models16
 - 5.4 Overview of ACSI services19
 - 5.5 Type definitions20
 - 5.5.1 General20
 - 5.5.2 Basic types22
 - 5.5.3 Common ACSI types23
- 6 SERVER class model28
 - 6.1 SERVER class definition28
 - 6.1.1 SERVER class syntax28
 - 6.1.2 SERVER class attributes28
 - 6.2 Server class services29
 - 6.2.1 Overview of directory and GetDefinition services29
 - 6.2.2 GetServerDirectory29
- 7 Application association model31
 - 7.1 Introduction31
 - 7.2 Concept of application associations31
 - 7.3 Access control31
 - 7.4 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model32
 - 7.4.1 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition32
 - 7.4.2 Two-party application association services34
 - 7.5 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class37
 - 7.5.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition37
- 8 LOGICAL-DEVICE class model38
 - 8.1 LOGICAL-DEVICE class definition38
 - 8.1.1 LOGICAL-DEVICE class syntax38
 - 8.1.2 LOGICAL-DEVICE class attributes38
 - 8.2 LOGICAL-DEVICE class services39
 - 8.2.1 GetLogicalDeviceDirectory39
- 9 LOGICAL-NODE class model40
 - 9.1 LOGICAL-NODE class definition40
 - 9.1.1 LOGICAL-NODE class syntax40
 - 9.1.2 LOGICAL-NODE class attributes40
 - 9.2 LOGICAL-NODE class services42
 - 9.2.1 Overview42
 - 9.2.2 GetLogicalNodeDirectory42

1	9.2.3	GetAllDataValues	43
2	10	DATA class model	45
3	10.1	General	45
4	10.2	DATA class definition	45
5	10.2.1	DATA class syntax	45
6	10.2.2	DATA class attributes	47
7	10.3	Relation of DATA, common DATA, and compatible DATA classes	54
8	10.4	DATA class services	54
9	10.4.1	General definitions and overview	54
10	10.4.2	GetDataValue	55
11	10.4.3	SetDataValue	56
12	10.4.4	GetDataDirectory	57
13	10.4.5	GetDataDefinition	58
14	11	DATA-SET class model	63
15	11.1	General	63
16	11.2	DATA-SET class definition	64
17	11.2.1	DATA-SET class syntax	64
18	11.2.2	DATA-SET class attributes	64
19	11.3	DATA-SET class services	65
20	11.3.1	Overview	65
21	11.3.2	GetDataSetValues	65
22	11.3.3	SetDataSetValues	66
23	11.3.4	CreateDataSet	67
24	11.3.5	DeleteDataSet	68
25	11.3.6	GetDataSetDirectory	69
26	12	Substitution model	70
27	13	SETTING-GROUP-CONTROL-BLOCK class model	72
28	13.1	General	72
29	13.2	SGCB class definition	73
30	13.2.1	SGCB class syntax	73
31	13.2.2	SGCB class attributes	74
32	13.3	SGCB class services	75
33	13.3.1	Overview	75
34	13.3.2	SelectActiveSG	75
35	13.3.3	SelectEditSG	76
36	13.3.4	SetSGValue	77
37	13.3.5	ConfirmEditSGValue	78
38	13.3.6	GetSGValue	78
39	13.3.7	GetSGCBValue	79
40	14	REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK class models	81
41	14.1	Overview	81
42	14.2	REPORT-CONTROL-BLOCK class model	83
43	14.2.1	Basic concepts	83
44	14.2.2	BUFFERED-REPORT-CONTROL-BLOCK (BRCB) class definition	83
45	14.2.3	BRCB class services	93
46	14.2.4	UNBUFFERED-REPORT-CONTROL-BLOCK (URCB) class definition	108
47	14.2.5	URCB class services	109
48	14.3	LOG-CONTROL-BLOCK class model	111

1	14.3.1	General	111
2	14.3.2	LCB class definition	112
3	14.3.3	LOG class definition.....	117
4	14.3.4	Procedures to generate the log entries.....	119
5	14.3.5	LOG services.....	119
6	15	Generic substation event class model (GSE)	123
7	15.1	Overview.....	123
8	15.2	GOOSE-CONTROL-BLOCK (GoCB) class	124
9	15.2.1	GoCB definition	124
10	15.2.2	GOOSE service Definitions	126
11	15.2.3	Generic object oriented substation event (GOOSE) message	131
12	15.3	Generic substation state event (GSSE) control block (GsCB)	172
13	15.3.1	GsCB class definition.....	172
14	15.3.2	Generic substation state event (GSSE) control block class attributes.....	173
15	15.3.3	GSSE service definitions.....	174
16	15.3.4	Generic substation state event (GSSE) message.....	179
17	16	Transmission of sampled value class model.....	134
18	16.1	Overview.....	134
19	16.2	Transmission of sampled values using multicast	135
20	16.2.1	MSVCB class definition.....	135
21	16.2.2	Multicast sampled value class services	137
22	16.3	Transmission of sampled values using unicast.....	140
23	16.3.1	USVCB class definition	140
24	16.3.2	Unicast sampled value services	142
25	16.4	Sampled value format.....	145
26	16.4.1	MsvID or UsvID.....	146
27	16.4.2	OptFlds	146
28	16.4.3	DatSet	146
29	16.4.4	Sample [1..n]	146
30	16.4.5	SmpCnt	146
31	16.4.6	RefrTm	147
32	16.4.7	ConfRev	147
33	16.4.8	SmpSynch	147
34	16.4.9	SmpRate	147
35	17	CONTROL class model.....	148
36	17.1	Introduction	148
37	17.2	Control with normal security	149
38	17.2.1	Direct control with normal security.....	149
39	17.2.2	SBO control with normal security.....	150
40	17.3	Control with enhanced security	151
41	17.3.1	Introduction	151
42	17.3.2	Direct control with enhanced security	151
43	17.3.3	SBO control with enhanced security	152
44	17.4	Time-activated operate	155
45	17.5	CONTROL class service definitions	156
46	17.5.1	Overview	156
47	17.5.2	Service parameter definition.....	157
48	17.5.3	Service specification	157

1	18	Time and time-synchronization model	162
2	18.1	General	162
3	18.2	External information	162
4	19	Naming conventions	163
5	19.1	Class naming and class specializations	163
6	19.2	Referencing an instance of a class	164
7	19.3	Scope	166
8	20	File transfer.....	168
9	20.1	File transfer model	168
10	20.1.1	FileName	168
11	20.1.2	FileSize [0..1]	168
12	20.1.3	LastModified	168
13	20.2	File services	169
14	20.2.1	GetFile	169
15	20.2.2	SetFile.....	169
16	20.2.3	DeleteFile	170
17	20.2.4	GetFileAttributeValues	171
18	Annex A	(normative) ACSI conformance statement	181
19	A.1	General.....	181
20	A.2	ACSI basic conformance statement	181
21	A.3	ACSI models conformance statement	182
22	A.4	ACSI service conformance statement	183
23	Bibliography	191
24	Index	192
25			
26	Figure 1	– Excerpt of conceptual model	15
27	Figure 2	– Basic conceptual class model of the ACSI	16
28	Figure 3	– Conceptual service model of the ACSI.....	18
29	Figure 4	– Data and control blocks (conceptionell model).....	20
30	Figure 5	– Data (CB) attribute type concept (example)	21
31	Figure 6	– Overview about GetDirectory and GetDefinition services	29
32	Figure 7	– Access views of a server.....	32
33	Figure 8	– Normal operation	33
34	Figure 9	– Aborting association.....	33
35	Figure 10	– Principle of multicast application association	37
36	Figure 11	– Class diagram of DATA and DataAttributeType.....	45
37	Figure 12	– Example of DATA.....	46
38	Figure 13	– Relation of TrgOp and Reporting	52
39	Figure 14	– Relation of DATA classes.....	54
40	Figure 15	– Excerpt of data class services	55
41	Figure 16	– Dynamic creation of data set instances.....	63
42	Figure 17	– Principles of substitution	71
43	Figure 18	– Basic model of the settings model	72
44	Figure 19	– Setting group state machine	74

1 Figure 20 – Basic building blocks for reporting and logging82

2 Figure 21 – BRCB state machine85

3 Figure 23 – Buffer time88

4 Figure 26 – Report example on the use of sequence number.....95

5 Figure 29 – Data set members and reporting.....98

6 Figure 30 – Report example.....100

7 Figure 31 – Log model overview111

8 Figure 32 – GoCB model123

9 Figure 33 – Specifics for GsCB model.....172

10 Figure 34 – Model for transmission of sampled values.....134

11 Figure 35 – Principle of the control model148

12 Figure 36 – State machine of direct control with normal security.....149

13 Figure 37 – Direct control with normal security149

14 Figure 38 – State machine of SBO control with normal security150

15 Figure 39 – State machine of direct control with enhanced security152

16 Figure 40 – State machine SBO control with enhanced security.....153

17 Figure 41 – Select before operate with enhanced security – positive case154

18 Figure 42 – Select before operate with enhanced security – negative case (no status

19 change).....154

20 Figure 43 – Time-activated operate.....156

21 Figure 44 – Time model and time synchronization (principle).....162

22 Figure 45 – Specializations.....163

23 Figure 46 – Object names and object reference166

24

25 Table 1 – ACSI classes.....19

26 Table 2 – BasicTypes22

27 Table 3 – ObjectName type.....23

28 Table 4 – ObjectReference type.....23

29 Table 5 – ServiceError type24

30 Table 6 – PACKED-LIST type25

31 Table 7 – TimeStamp type25

32 Table 8 – TimeQuality definition.....26

33 Table 9 – TimeAccuracy26

34 Table 10 – TriggerConditions type27

35 Table 11 – SERVER class definition28

36 Table 12 – TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class definition33

37 Table 13 – MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition.....37

38 Table 14 – LOGICAL-DEVICE (LD) class definition38

39 Table 15 – LOGICAL-NODE (LN) class definition40

40 Table 16 – DATA class definition47

41 Table 17 – DAType definition48

42 Table 18 – Functional constraint values50

43 Table 19 – TrgOp51

1	Table 20 – COMMON-DATA class definition.....	53
2	Table 21 – DATA-SET (DS) class definition.....	64
3	Table 22 – SGCB class definition.....	73
4	Table 23 – BRCB class definition.....	84
5	Table 24 – Report format specification.....	94
6	Table 25 – ReasonForInclusion.....	99
7	Table 25 – URCB class definition.....	108
8	Table 27 – LCB class definition.....	112
9	Table 28 – LOG class definition.....	117
10	Table 29 – GOOSE control block class definition.....	124
11	Table 30 – GOOSE message definition.....	132
12	Table 31 – GSSE control block class definition.....	173
13	Table 32 – GSSE message definition.....	179
14	Table 33 – MSVCB class definition.....	135
15	Table 34 – USVCB class definition.....	140
16	Table 35 – Sampled value (SV) format definition.....	146
17	Table 35 – Control services.....	156
18	Table 37 – Additional cause diagnosis definition.....	Fehler! Textmarke nicht definiert.
19	Table 38 – AddCause semantic.....	Fehler! Textmarke nicht definiert.
20	Table 39 – TimeActivatedOperate response definition.....	Fehler! Textmarke nicht definiert.
21	Table 40 – List of ObjectReferences.....	164
22	Table 41 – FILE class definition.....	168
23	Table A.1 – Basic conformance statement.....	181
24	Table A.2 – ACSI models conformance statement.....	182
25	Table A.3 – ACSI service conformance statement.....	183
26	Table A.3 (<i>continued</i>).....	184
27	Table A.3 (<i>continued</i>).....	185
28	Table A.3 (<i>continued</i>).....	186

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMUNICATION NETWORKS AND SYSTEMS
FOR POWER UTILITY AUTOMATION –

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

FOREWORD

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

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

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

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

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

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

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

- 1 Part 7-2: Basic information and communication structure for substation and feeder equip-
2 ment – Abstract communication service interface (ACSI)
- 3 Part 7-3: Basic information and communication structure for substation and feeder equip-
4 ment – Common data classes
- 5 Part 7-4: Basic information and communication structure for substation and feeder equip-
6 ment – Compatible logical node classes and data classes
- 7 Part 8-1: Specific communication service mapping (SCSM) – Mappings to MMS (ISO/IEC 9506-1
8 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
- 16

1 INTRODUCTION

2 This document is part of a set of definitions which details a layered utility communication ar-
3 chitecture. This architecture has been chosen to provide abstract definitions of classes and
4 services such that the definitions are independent of specific protocol stacks, implementa-
5 tions, 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
9 services 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 de-
13 vices. The ACSI has been defined so as to be independent of the underlying communication
14 systems. Specific communication service mappings¹ (SCSM) are specified in part 8-x and
15 part 9-x of this 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 imple-
22 ment 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
24 breaker) or a real function that are visible and accessible over a communication network are modelled. This ab-
25 straction 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
28 SCSMs.

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

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

33
34 **Editor's Note:**
35 This document is based on the experiences made after the publication of the first Edition of
36 this standard. The crucial changes made in this CD are according to the Tissues on part IEC
37 61850-7-2 (Technical Issues) posted at the Tissue database (www.tissue.iec61850.com).

38
39

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
10 semantics

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
18 entities 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 gener-
23 ation, 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
31 state 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

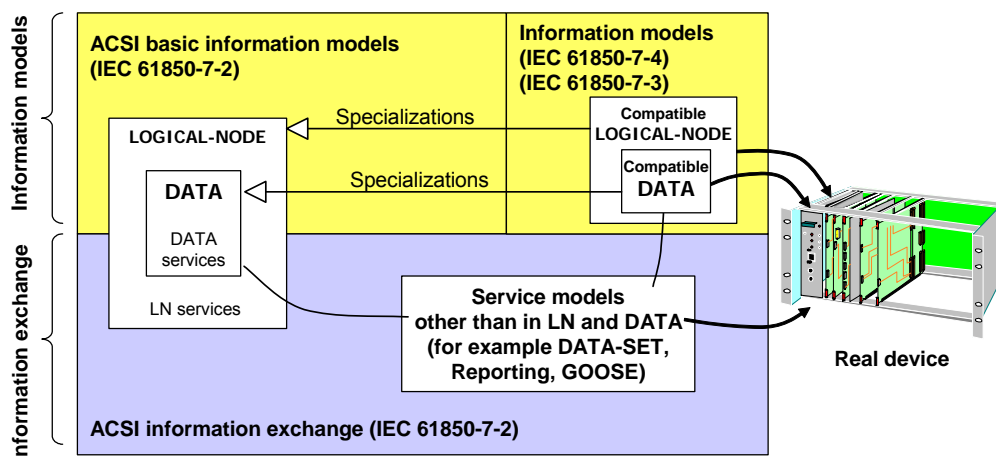
1 5 ACSI overview and basic concepts

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 descrip-
 10 tive point of view, the two aspects are separated to some degree (see the excerpt shown in
 11 Figure 1). The common models (for example, logical node and data classes including their
 12 services) are applied in IEC 61850-7-3 and IEC 61850-7-4 to define many specialized infor-
 13 mation models for utility automation models, or in IEC 61400-25-2 to define specialized mod-
 14 els for 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
- 25 are 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 val-
 28 ues).

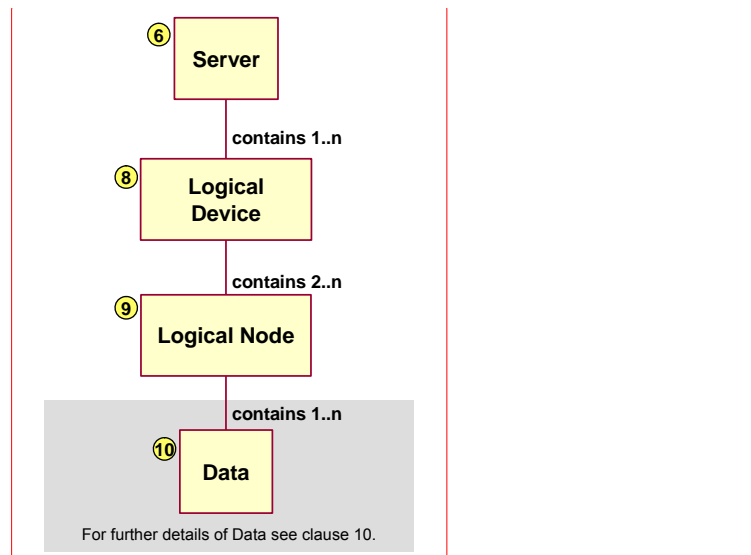
- 29 b) **Logical device (LD)** – contains the information produced and consumed by a group of
- 30 domain-specific application functions; functions are defined as logical nodes.

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

3 d) **Data** – provide means to define typed information, for example, position of a switch with
 4 quality information and timestamp, contained in logical nodes.

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

7 NOTE 2 The classes are major building blocks that provide the framework for utility automation device models.
 8 Additional details on the modelling and relations between IEC 61850-7-4, IEC 61850-7-3, and this part of IEC
 9 61850 can be found in IEC 61850-7-1.



Kommentar [KS1]: Tissue 390. Changed 1..* to 1..n. Updated LN multiplicity from 1 to 2. Removed DataAttribute. Simplified model notation used

10
 11 **Figure 2 – Basic conceptual class model of the ACSI**

12 NOTE 3 The numbers in the circles indicate the respective clauses in this part of IEC 61850.

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

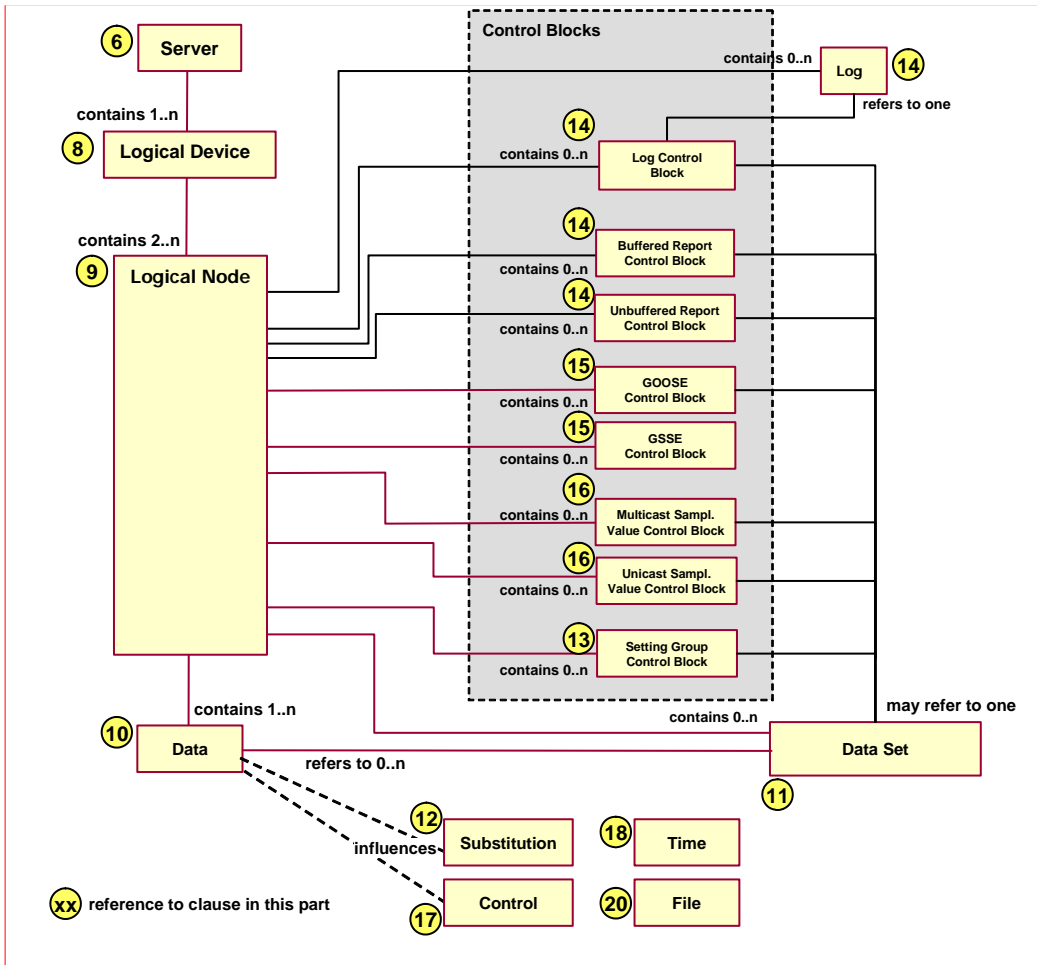
Kommentar [KS2]: Tissue 390

15 EXAMPLE In an implementation the logical device, logical node, data, and data attribute have each an object
 16 name (instance name) which is a unique name among classes of the same container to which they belong. In addition,
 17 each of the four has an ObjectReference (path name) which is a concatenation of all object names from each
 18 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

19
 20 **5.3 Overview of the other service models**
 21 In addition to the models listed above, the ACSI comprises the following models that provide
 22 services operating on data, data attributes, and data sets.

- 1 a) **Data Set** – permits the grouping of data and data attributes. Used for direct access and
2 for reporting and logging.
- 3 b) **Substitution** – supports replacement of a process value by another value.
- 4 c) **Setting group control** – defines how to switch from one set of setting values to another
5 one and how to edit setting groups.
- 6 d) **Report control and logging** – describe the conditions for generating reports and logs
7 based on parameters set by the client. Reports may be triggered by changes of process
8 data values (for example, state change or dead band) or by quality changes. Logs can be
9 queried for later retrieval. Reports may be sent immediately or deferred. Reports provide
10 change-of-state and sequence-of-events information exchange.
- 11 e) **Control blocks for generic substation event (GSE)** – supports a fast and reliable sys-
12 tem-wide distribution of input and output data values; peer-to-peer exchange of IED bi-
13 nary status information, for example, a trip signal.
- 14 f) **Control blocks for transmission of sampled values** – fast and cyclic transfer of sam-
15 ples, for example, of instrument transformers.
- 16 g) **Control** – describes the services to control, for example, devices.
- 17 h) **Time and time synchronization** – provides the time base for the device and system.
- 18 i) **File transfer** – defines the exchange of large data blocks such as programs.
- 19 An overview of the conceptual service model of the ACSI is shown in Figure 3.



Kommentar [KS3]: Tissue 192, 384
 LN multiplicity changed from 1 to 2.
 Changed "*" to "n"
 #384: LOG associated with LN class. Allow more than one LOG (done).
 #387: changed multiplicity of GSE control blocks to 0..n.
 #388 Changed names to attributes names not Class names. Here it is an introduction only.
 Simplified model notation used.
 LNN0 has been removed according to Editros' meeting in Zug

Figure 3 – Conceptual service model of the ACSI

1
 2
 3
 4
 5
 6
 7
 8
 9
 10
 11
 12
 13
 14
 15
 16

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

NOTE 2 The class diagrams are conceptual. Details are defined in the respective clauses. Comprehensive diagrams are contained in IEC 61850-7-1. The data class may be defined recursively. The operations for substitution and control are restricted to the lowest level in the data class. The DataAttributes may be defined recursively as well.

The logical node is one of the major building blocks that have associations to most of the other information exchange models, for example, report control, log control, and setting control. Control block classes shall only be available in a Logical Node instance if support is stated in the definition of a compatible LN class.

Any other information exchange service model, for example, report control, log control, and setting control shall inherit the ObjectName and ObjectReference as depicted in Figure 2.

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

Kommentar [KS4]: According to editors' meeting in Zug

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 DataSetValues 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 (Clause 4, 5)</u> 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**

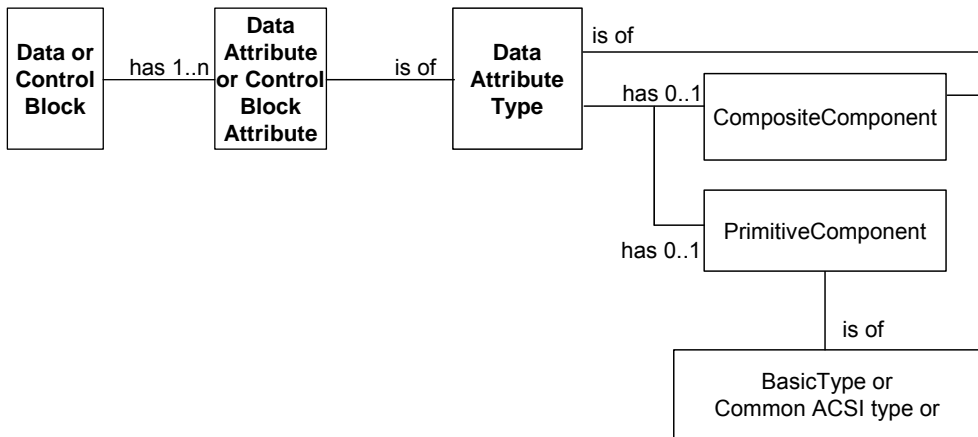
2 **5.5.1 General**

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



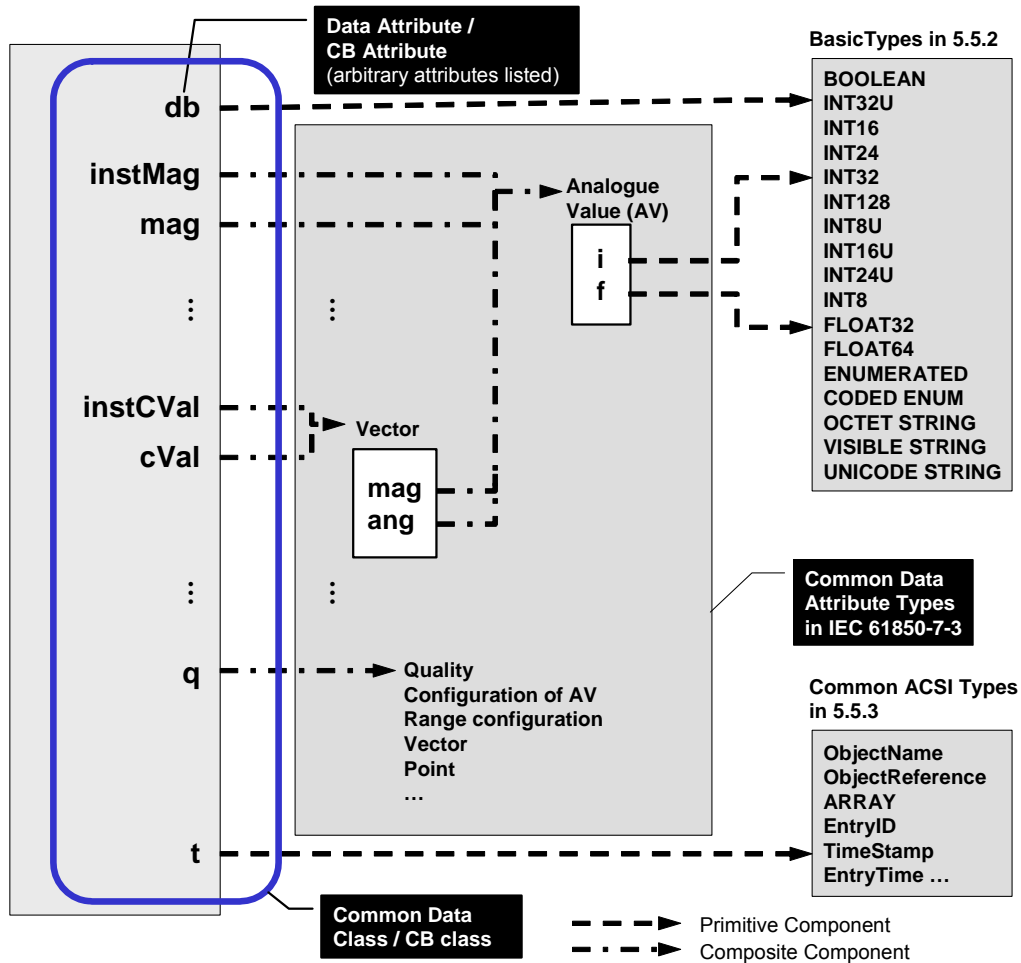
15
16 **Figure 4 – Data and control blocks (conceptionell model)**

17 The data (e.g., measured value) shown on the left side has one or more data attributes (current value, time stamp, quality information). Each data attribute is of a specific type (the data attribute type). The data attribute type can be primitive or can comprise structured components. The primitive component can be of a BasicType (see 5.5.2) or a common ACSI type (see 5.5.3), The composite component can, e.g., be a vector comprising a magnitude and a angle value, 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 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 61850-7-3).



1

2

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

3
4
5
6
7

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

8
9
10
11

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

12
13
14
15
16

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

1 **5.5.2 Basic types**

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

3 **Table 2 – BasicTypes**

BasicTypes			
Name	Value range	Remark	Used by
BOOLEAN	0,1		IEC 61850-7-3 IEC 61850-7-2
INT8	-128 to 127		IEC 61850-7-3 IEC 61850-7-2
INT16	-32 768 to 32 767		IEC 61850-7-3 IEC 61850-7-2
INT24	-8 388 608 to 8 388 607	for TimeStamp type	IEC 61850-7-2
INT32	-2 147 483 648 to 2 147 483 647		IEC 61850-7-3 IEC 61850-7-2
INT128	-2**127 to (2**127)-1	Required for counters	IEC 61850-7-3
INT8U	Unsigned integer, 0 to 255		IEC 61850-7-3 IEC 61850-7-2
INT16U	Unsigned integer, 0 to 65 535		IEC 61850-7-3 IEC 61850-7-2
INT24U	Unsigned integer, 0 to 16 777 215		IEC 61850-7-2
INT32U	Unsigned integer, 0 to 4 294 967 295		IEC 61850-7-3 IEC 61850-7-2
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

Kommentar [KS5]: Tissue #392 and editors

Kommentar [KS6]: Tissue #392 and editors

^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 exam-
4 ple, report control blocks) defined in this part of IEC 61850. The common ACSI types may
5 also be 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

Kommentar [KS7]: New length required because of IEC 61850-8-1: 2004 clause G.3.2

Kommentar [KS8]: According to the editors' meeting in Zug.

10 The constraints defined in clause 19 on the use of the type ObjectReference shall be applied.

11 5.5.3.3 ObjectReference

12 Instances of classes in the hierarchical information model (ACSI class hierarchy of logical
13 device, 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

Kommentar [KS9]: New length required because of IEC 61850-8-1: 2004 clause G.3.2 64 + "/" + 64 char maximum

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
20 instance of a logical node (**LNName**). The "." shall separate the further names in the hierar-
21 chy. The "[.]" shall indicate an option. The inner square bracket "[. ...]" shall indicate further
22 names of recursively nested definitions.

Kommentar [KS10]: Added the "." To differentiate this from the ArrayElement number in the DataAttributeReference definition [ArrayElement]

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 The constraints defined in clause 19 on the use of the type ObjectReference shall be applied.

Kommentar [KS11]: According to the editors' meeting in Zug.

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.

Kommentar [KS12]:
#142; needs to be added to the
corresponding control blocks.

4 **5.5.3.5 ARRAY type**

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

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

Kommentar [KS13]: Note
that the start value of 0 means
that the array has m+1 array
elements
[] removed according to editors'
meeting in Zug
P = 1. added

- 7 with $m \geq 0$
- 8 $p =$ 1. Common Data Class (CDC, according to IEC 61850-7-3) that does not
- 9 contain an ARRAY type
- 10 2. Common data attribute type or
- 11 3. BasicType or
- 12 4. Common ACSI type (except ARRAY type)

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

15 The Common Data Class used as the the type "p" of the ARRAY may be functionally con-
16 straint (see 10.2.2.4.2).

Kommentar [KS14]: Added
according to editors' meeting in
Zug.

17 The ARRAY type shall be applied for DataAttributes only.

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

Kommentar [KS15]: Tis-
sue #51. The definition takes
into account all inputs as per
2007-01-03

19 **5.5.3.6 ServiceError type**

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

22 **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

23
24 Additional **ServiceError** values for negative service responses (originated in the application,
25 for example, additional cause diagnosis for control-related services) shall be as specified in
26 the appropriate service models.

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

28 **5.5.3.7 EntryID type**

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

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

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

4 5.5.3.8 Packed list type

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

6 **Table 6 – PACKED-LIST type**

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

7

8 5.5.3.9 TimeStamp type

9 5.5.3.9.1 General

10 The relation between a time stamp value, the synchronization of an internal time with an ex-
11 ternal time source (for example, UTC time), and other time-model-related information are de-
12 fined in Clause 18.

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

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

16 5.5.3.9.2 TimeStamp syntax

17 The **TimeStamp** type shall represent a UTC time with the epoch of midnight (00:00:00) of
18 1970-01-01 specified in Table 7.

19 **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 $b_i \cdot 2^{30-i}$; Order = b0, b1, b2, b3, ...	M
TimeQuality	TimeQuality		M

Kommentar [KS16]: Tissue #325

20

21 5.5.3.9.3 TimeStamp attributes

22 5.5.3.9.3.1 SecondSinceEpoch

23 The **SecondSinceEpoch** shall be the interval in seconds continuously counted from the ep-
24 och 1970-01-01 00:00:00 UTC.

25 NOTE SecondSinceEpoch corresponds with the Unix epoch.

26 5.5.3.9.3.2 FractionOfSecond

27 The attribute **FractionOfSecond** shall be the fraction of the current second when the value
28 of the **TimeStamp** has been determined. The fraction of second shall be calculated as (SUM
29 from l = 0 to 23 of $b_l \cdot 2^{30-l}$ s).

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

3 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 $1/2^{*2}$ (=
 4 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
 5 by an IED is outside the scope of this standard.

6 **5.5.3.9.3.3 TimeQuality**

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

9 **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

10

11 **LeapSecondsKnown:** The value TRUE of the attribute **LeapSecondsKnown** shall indicate
 12 that the value for SecondSinceEpoch takes into account all leap seconds occurred. If it is
 13 FALSE then the value does not take into account the leap seconds that occurred before the
 14 initialization of the time source of the device.

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

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

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

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

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

24 **Table 9 – TimeAccuracy**

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

25

1 **5.5.3.10 EntryTime type**

2 The type **EntryTime** shall represent the time and date as applied internally for the communi-
 3 cation, reporting, logging, and subsystem as specified by a SCSM.

4 The time base for **EntryTime** shall be GMT. The epoch for **EntryTime** shall be 01. January
 5 1984 (MJD 40 587).

Kommentar [KS17]: Re-
 quired for harmonization with
 edition 2 in 8-1. Clarifies that
 EntryTime is NOT local time.
 Editors' meeting

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

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

10 **5.5.3.11 TriggerConditions type**

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

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

14 **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

15
 16 The **TriggerOption (TrgOp)** shall be used in the definition of **DataAttributes** to indicate on
 17 which change/update the value of an instance of a **DataAttribute** may be reported or
 18 logged.

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

20 **5.5.3.12 CtxInt type**

Kommentar [KS18]: Tis-
 sue #234 / #146
 Also official comment CH01

21 Attributes of this type can either be of the basic type **INT32** or of the basic type
 22 **ENUMERATED**. This depends on the semantic of the data attribute as it is defined with the
 23 specification of the logical node class and data class.

24 Value range and mapping depend on the use and SCSM.

25 **EXAMPLE** CDC INS uses CtxUnt ... tbd

1 **6 SERVER class model**

2 **6.1 SERVER class definition**

3 **6.1.1 SERVER class syntax**

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

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

8 **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	
MCAppAssociation [0..n]	MULTICAST-APPLICATION-ASSOCIATION	
Services		
GetServerDirectory		

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

14 **6.1.2 SERVER class attributes**

15 **6.1.2.1 ServiceAccessPoint [1..n]**

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

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

21 **6.1.2.2 LogicalDevice [1..n]**

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

23 **6.1.2.3 File [0..n]**

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

25 **6.1.2.4 TPAppAssociation [0..n] – two-party application association**

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

28 NOTE Details can be found in Clause 7.

Kommentar [KS19]: Consistency issue; consistent with table 11.

1 **6.1.2.5 MCAAppAssociation [0..n] – multicast application association**

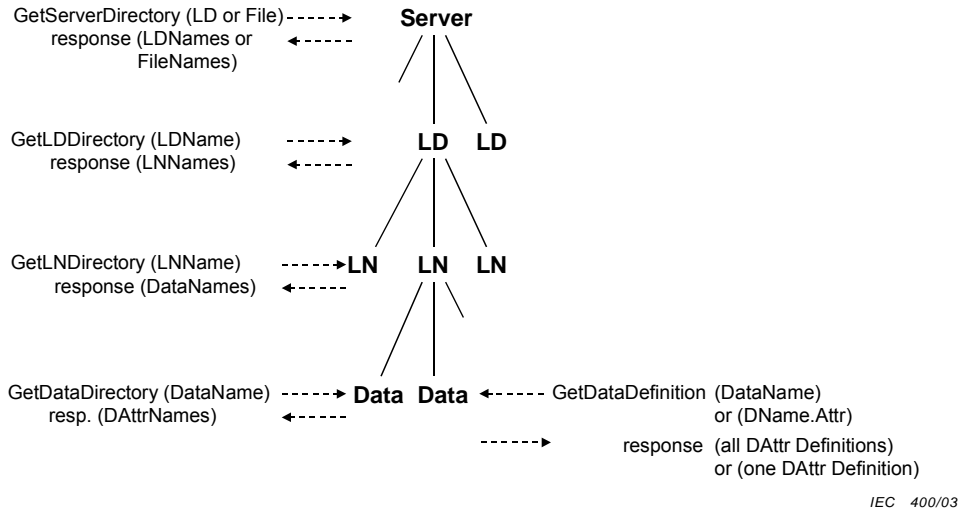
2 The attribute **MCAAppAssociation** shall identify a subscriber with which a **SERVER** (pub-
3 lisher) maintains a multicast application association.

4 NOTE Details can be found in Clause 7.

5 **6.2 Server class services**

6 **6.2.1 Overview of directory and GetDefinition services**

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



9

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

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

14 **6.2.2 GetServerDirectory**

15 **6.2.2.1 GetServerDirectory parameter table**

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

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

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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17

6.2.2.2 Request

6.2.2.2.1 ObjectClass

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

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

6.2.2.3 Response+

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

6.2.2.3.1 Reference [0..n]

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

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

6.2.2.4 Response-

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

Kommentar [KS20]: Tissue # 409. The ObjectClass parameter of the service does not carry a real CLASS! it has only an identification for which kind of classes the service is looking for.

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 A new work item on the definition of access control roles has been pulished: Role-Based Ac-
11 ccess Control for Power systems management (57/914/NP). The results of that project is in-
12 tended to be used in IEC 61850 when available.

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

18 7.2 Concept of application associations

19 The application association model defines

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

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

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

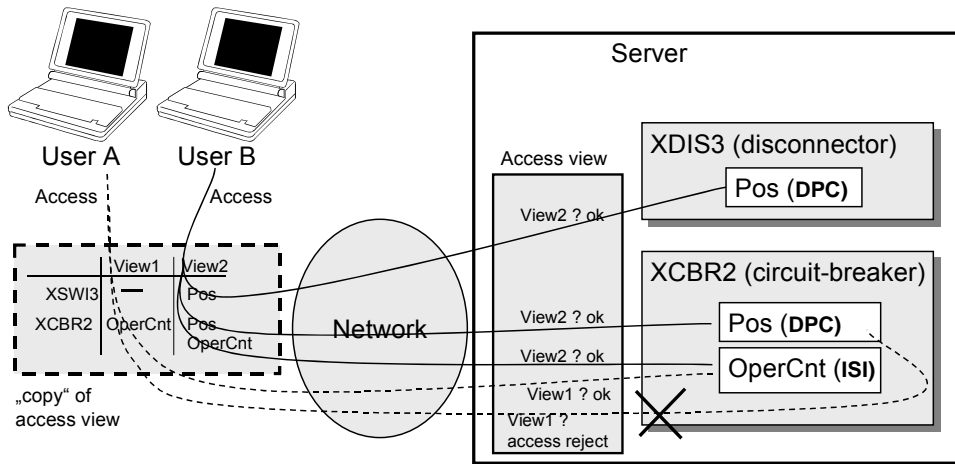
29 NOTE The details of an application association model are defined in the SCSMs. The following descriptions pro-
30 vide a conceptual model of the application associations between devices.

31 7.3 Access control

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

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

Kommentar [KS21]: No changes so far. Need to discuss how the NWIP N 914 fits into this description.



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
 7 device, thus providing **access restriction to any user** who tries to access the instances. Inde-
 8 pendent of the implementation in the device, additional access restriction may be imple-
 9 mented at the 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 identi-
 14 fied by authentication parameters passed to the server when establishing the association
 15 with the server (two-party application association) or when sending information over mul-
 16 ticast 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
 18 the 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.

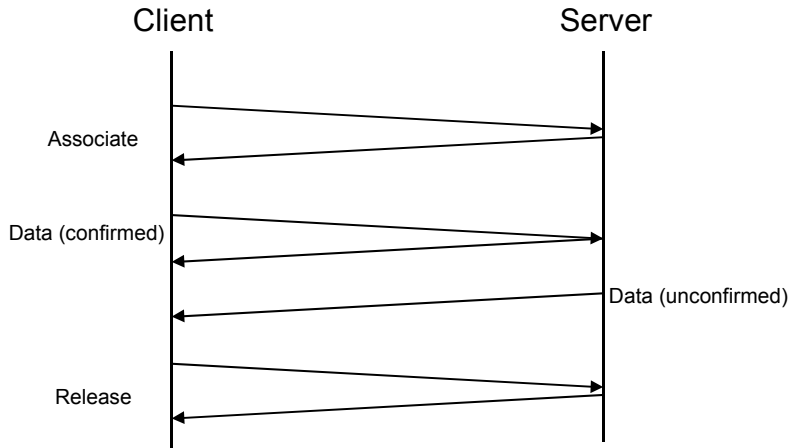
21 7.4 TWO-PARTY-APPLICATION-ASSOCIATION (TPAA) class model

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
 27 association relies provides measures to notify reasons for non-deliverance of information in
 28 due time. End-to-end flow control means that sources of information do not send more infor-
 29 mation than the destination can buffer.

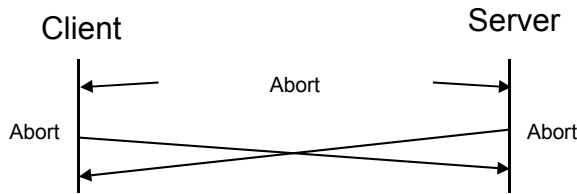
1 The services for associate, data exchange, and association release of the two-party applica-
 2 tion association class is depicted in Figure 8.



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
ServerAccessPointRefer- ence
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 appli-
24 cation 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
5 be 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
16 **ServiceError** 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 – no
21 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
27 the 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
31 remote 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
21 successful.

Kommentar [KS22]: #498

22 **7.4.2.4.4 Response-**

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

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

Kommentar [KS23]: #498

1 **7.5 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class**

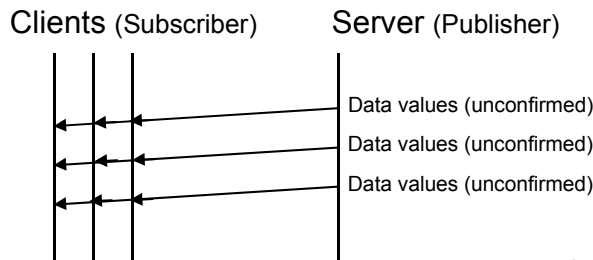
2 **7.5.1 MULTICAST-APPLICATION-ASSOCIATION (MCAA) class definition**

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
 7 information 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 dupli-
 10 cated 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
 22 to 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"
 28 that carries association parameters and data. The "application association" ceases as soon as the service has
 29 been 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		

Kommentar [KS24]: Tissue 369

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
13 a system.

14 **Editor's Note:** We have decided to replace system by subnetwork. This causes the same problem: Subnetwork is
15 not defined in 61850. In computer networks that use the Internet Protocol, a subnetwork or subnet is a range of
16 logical addresses within the address space that is assigned to an organization. We need to further discuss the
17 scope.

Kommentar [KS25]: We need to define what a system is! Any proposal? What is with more than one LD? Can we have more than one LD0, e.g., if we have more than one proxy? Second: Uniquely means: in "normal" IEDs. In Proxies we have "copies" of the LDs ... So we may have at least two LDx: one in a non-proxy IED and one in a proxy IED.

18 **8.1.2.2 LDRef – logical device ObjectReference**

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

LDName

20

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

23 **8.1.2.3 LogicalNode [2..n]**

24 The attribute **LogicalNode [2..n]** shall identify all logical nodes of type the **LOGICAL-NODE**
25 that are contained in a **LOGICAL-DEVICE**.

Kommentar [KS26]: Tissue 369; should the Attribute be plural? LogicalNodes?

Kommentar [KS27]: To make the definition consistent. "n" was not defined before. Even it is obvious what it means, it is required to define it explicitly.

26 Each **LOGICAL-DEVICE** shall have one and only one **LOGICAL-NODE-ZERO (LLNO)**,
27 one and only one **LOGICAL-NODE-PHYSICAL-DEVICE (LPHD)**, and it may have no or
28 several other **LOGICAL-NODEs**.

29 NOTE The **LLNO**, **LPHD**, and other logical nodes are defined in IEC 61850-7-4 for utility 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 Ob-
5 jectReferences of all **LOGICAL-NODEs** made visible and thus accessible to the requesting
6 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
8 the view concept).

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

Kommentar [KS28]: Tissue 369

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-**
13 **DEVICE**.

14 8.2.1.3 Response+

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

17 8.2.1.3.1 LNReference [2..n] – logical node ObjectReference

18 The parameter **LNReference** [2..n] shall contain all ObjectReferences **LNRef** of a
19 **LOGICAL-NODE** from the referenced **LOGICAL-DEVICE**.

20 Two references returned shall be a **LOGICAL-NODE-ZERO (LLN0)** and a **LOGICAL-**
21 **NODE-PHYSICAL-DEVICE (LPHD)**.

Kommentar [KS29]: To make the definition consistent.

Kommentar [KS30]: LNReference without [2..n] does not define that the service should return ALL names of LNs! It is not said that "n" is the number of LNs contained in the LD.

Kommentar [KS31]: To make the definition consistent.

22 8.2.1.4 Response–

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

Kommentar [KS32]: Make the definition consistent with the rest.

25

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	
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.		
SettingGroupControlBlock [0..1]	SGCB	
Log [0..n]	LOG	
LogControlBlock [0..n]	LCB	
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.		

Kommentar [KS33]: According to editors' meeting in Zug

Kommentar [KS34]: Tissue #384: LOG class associated with LN class. Changed multiplicity from [0..1] to [0..n] (needs to be discussed and decided)

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

Kommentar [KS35]: According to the decision on the new title made in Arnhem (2006-10).

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 con-
14 tained 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
17 contained 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-**
20 **NODE**.

Kommentar [KS36]: Tissue #384: changed multiplicity to [0..1] to [0..n].

Kommentar [KS37]: Changed from LLN= to LOGICAL-NODE

21 **9.1.2.8 LogControlBlock [0..n]**

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

24 **9.1.2.9 SettingGroupControlBlock [0..1]**

25 The attribute **SettingGroupControl** shall identify the **SGCB** (see Clause 13) that is con-
26 tained in a **LLNO**.

27 **9.1.2.10 GOOSEControlBlock [0..n]**

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

30 **9.1.2.11 GSSEControlBlock [0..n]**

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

1 **9.1.2.12 MulticastSampledValueControlBlock [0..n]**

2 The attribute **MulticastSampledValueControlBlock** shall identify a **MSVCB** (see 16.2) that
3 is 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 Objec-
14 tReferences of all instances of a requested class made visible and thus accessible
15 to the requesting client by the referenced **LOGICAL-NODE**.

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

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

18

19 **9.2.2.2 Request**

20 **9.2.2.2.1 LNReference**

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

23 **9.2.2.2.2 ACSIClass**

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

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

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

Kommentar [KS38]: Tissue # 410. The ObjectClass parameter of the service does not carry a real CLASS! it has only an identification for which kind of classes the service is looking for.

1 **9.2.2.3 Response+**

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

4 **InstanceName [0..n]**

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

7 The parameter **InstanceName [0..n]** shall contain all **ObjectNames** of the requested class.

Kommentar [KS39]: To make the definition consistent.

8 If the ACSI class model requested equals DATA:

9 The parameter **InstanceName [0..n]** shall contain all **ObjectNames** of the highest level of
10 **DATA**.

11 NOTE: If the **DATA** class contains a **DATA** class (e.g. in the CDC WYE) the **ObjectNames** of the underlying **DATA**
12 class may be retrieved by the service **GetDataDictionary** (10.4.4).

Kommentar [KS40]: According to the editors' meeting in Zug.

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

15 **9.2.2.4 Response-**

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

18 **9.2.3 GetAllDataValues**

Kommentar [KS41]: The use of FCCB for GetAllData-Values needs to be discussed! Do we want to separate DATA and CB completely? Or do we use this service also to get all values of CBs? Thus using the FCD, FCDA and FCCB?

19 **9.2.3.1 GetAllDataValues parameter table**

20 A client shall use the **GetAllDataValues** service to retrieve all **DataAttribute** values (hav-
21 ing the same **FunctionalConstraint**) of all **DATA** made visible and thus accessible to the
22 requesting client by the referenced **LOGICAL-NODE**.

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

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

25

1 **9.2.3.2 Request**

2 **9.2.3.2.1 LNReference**

3 The parameter **LNReference** shall contain the ObjectReference **LNRef** of the **LOGICAL-**
4 **NODE**.

5 **9.2.3.2.2 FunctionalConstraint [0..1]**

6 The parameter **FunctionalConstraint (FCCB)** shall contain the functional constraint pa-
7 rameter (**FC**) to filter the respective **DataAttributes** of all **DATA** contained in the **LOGICAL-**
8 **NODE**. The **FC** shall be as defined in 10.2.2.4.2.

9 **9.2.3.3 Response+**

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

12 **9.2.3.3.1 DataAttributeReference [1..n]**

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

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

17 **9.2.3.3.2 DataAttributeValue [1..n]**

18 The parameter **DataAttributeValue** shall contain the value of a **DataAttribute** of the **DATA**
19 contained in the referenced **LOGICAL-NODE**. If the parameter **FunctionalConstraint** is
20 present in the service request then only values of those **DataAttributes** that have the **Func-**
21 **tionalConstraint** as given in the service request shall be returned.

Kommentar [KS42]: Defini-
tion revised for consistency with
paragraph 9.2.3.3.1

22 **9.2.3.4 Response-**

23 The parameter **Response-** shall indicate that the service request failed. The appropriate
24 **ServiceError** 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, **OiFil** for oil filtration. The composi-
 7 tion of **DATA** in IEC 61850-7-4 is based on common templates (the common **DATA** classes,
 8 **CDC**) specified in IEC 61850-7-3. The concept of **DATA** classes is introduced in this clause.
 9 Any set of **DATA** (or parts of **DATA**) instances may be grouped to build **DATA-SET** instances
 10 applying the **CreateDataSet** service. **DATA-SET** instances can, for example, be written
 11 (**SetDataSetValues**) or read (**GetDataSetValues**)

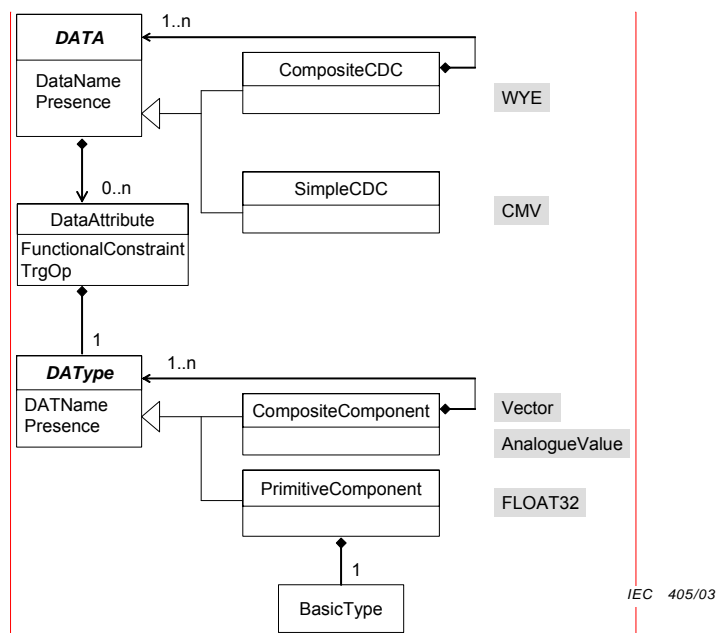
12 NOTE 1 The consequences of setting values to instances of **DATA** is outside this part of IEC 61850. IEC 61850-7-3 and
 13 IEC 61850-7-4 define many utility-domain-specific **DATA** classes. These definitions provide information on the ac-
 14 tions to be taken by the receiving application, for example, 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. Uncon-
 19 trolled spontaneous transmission may congest the network. Services for a controlled reporting are specified in
 20 Clause 14.

21 10.2 DATA class definition

22 10.2.1 DATA class syntax

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



Kommentar [KS43]: #399;
 diagram revised

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

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

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

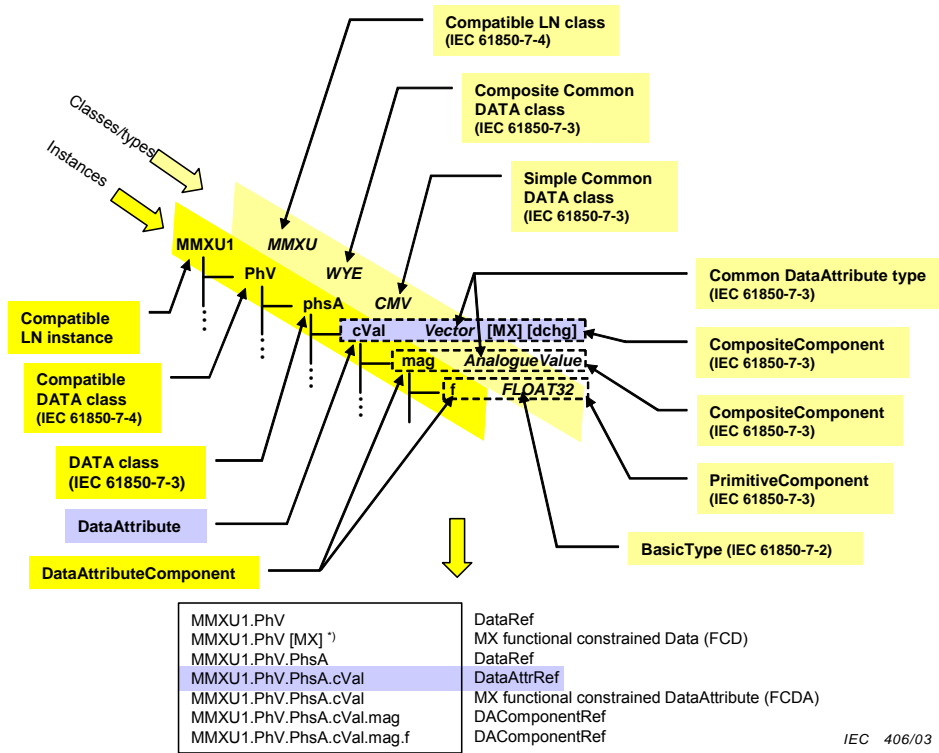
3 NOTE 2 The **DATA** class is an abstract class that is an auxiliary means to construct the primitive and composite
 4 common data classes.

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**
 7 (simple common data class) and **CompositeCDC** (composite common data class). **Simple**
 8 **leCDC** shall have a **DataName**, a **Presence**, and **DataAttributes** (for example, **DataName**
 9 = **phsA**, **Presence** = **Mandatory**, and **DataAttribute** = **cVal**). The **CompositeCDC** is con-
 10 structed by one or more **SimpleCDC** and/or **DataAttributes** (for example, **CDC WYE** com-
 11 prising a **SimpleCDC** **CMV**, etc.).

12 The **DataType** 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
 19 **DataAttribute** has additionally the functional constraint **FC = MX** (measurand) and the trigger
 20 option **TrgOp = dchg** (data-change).



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

Figure 12 – Example of DATA

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
3 of 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 **DAType** shall be as shown in Figure 11.

8 The table notation does not easily show the inheritance. Therefore the class diagram in
9 Figure 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..2]	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		

Kommentar [KS44]: Official comment JP1

Kommentar [KS45]: Tissue of editor. Can be zero or one only per data attribute.

11

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

15 NOTE 5 The structure of a **DATA class** is recursive since a **CompositeCDC** is also of type **DATA class**. The
16 level of recursion may be restricted by a SCSM, so the number of levels of recursion of **CompositeCDCs** is normally
17 not 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 expected
19 as long as they are referenced as members of a **DATA-SET**. A system has to take special measures to ensure
20 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
5 a **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.		

Kommentar [KS46]: #501

16

17 **DATName – data attribute type name**

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

The **DATName** (if **DataAttribute** is not nested) or the **DATName** of the first level (if **DataAttribute** 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[.DAComponentName[. ...]]

7 The **ObjectReference DataAttributeReference** shall be one of the two definitions:

8 If the **DataAttribute** is not an **Array element**: The **ObjectReference DataAttributeReference**
 9 shall be:

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

10 If the **DataAttribute** is an **Array element**: The **ObjectReference DataAttributeReference**
 11 shall be:

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

Kommentar [KS47]: The two cases are differentiated because: the **NumArrayElement** cannot be optional, if not given and the data is an array, it would mean ALL array elements. That is not possible, because. Either we have a non-Array **DataAttribute** which just has a **DAName**, or we have an **Array**. Each **Array element** is a **DataAttribute**.

Kommentar [KS48]: According to decision of San Diego meeting 2007-02

12
 13 **NOTE 3** Nesting depends on the concrete definition of a **DATA** class and **DAType** class.

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

16 The **NumArrayElement** (value between 0 and m according to the instance of the array) shall
 17 be the array element number in case a single array element is to be referenced.

Kommentar [KS49]: According to decision of San Diego meeting 2007-02

18 **Presence**

19 The attribute **Presence** of type **BOOLEAN** shall define if a **DataAttribute** is mandatory
 20 (**Presence = TRUE**) or optional (**Presence = FALSE**).

21 **NOTE** The **Presence** attribute is a conceptual attribute only. The common data classes as defined, e.g., in IEC
 22 61850-7-3 use a column "M/O/C" (mandatory / optional / conditional) to represent if a data attribute or common
 23 data attribute shall be present in an implementation.

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

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

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

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

5 **10.2.2.4.2 FunctionalConstraint for DATA**

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

Kommentar [KS50]: Removed "[1..2]" ... shall be available for all DataAttributes.
Kommentar [KS51]: Added "for DATA" because we have to differentiate these from the control block functional constraints.

10 The **FunctionalConstraint (FC)** shall be a property of the **DataAttribute** characterizing the specific use of the **DataAttribute**. The **FunctionalConstraint** is used in the definition of **DATA** (contained in **LOGICAL-NODES**).

Kommentar [KS52]: Removed "and of the various control blocks (for example, BRBC). Most attributes of control blocks have a **FunctionalConstraint** property." because we have CB specific definition in new clause 11.

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

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

18 **Table 18 – Functional constraint values**

FunctionalConstraint values			
FC	Semantic	Services allowed	Initial values/storage/ explanation
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
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
CO	Control	DataAttribute shall represent a control information whose value may be operated (control model) and read	N.a.
SP	Setpoint	DataAttribute shall represent a setting parameter information 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
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
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
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
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 cur-	Initial value of the DataAttribute shall be as configured; value shall be non-volatile

Kommentar [KS53]: #220; changes as proposed. "one" "whose value" removed

FunctionalConstraint values			
FC	Semantic	Services allowed	Initial values/storage/ explanation
		rent active value (for details see 13). Values the of DataAttributes with FC=SG shall not be writeable	
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 processed
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
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 for the common data class **single-point status (SPS)** according to
3 IEC 61850-7-3 has the following **DataAttributes**: **stVal** (status 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
8 a **DATA**) that may cause a report to be sent or a log entry to be stored into a log (report
9 model; see Clause 14). The services associated with the **TrgOpt** shall be as specified in Ta-
10 ble 19.

11 Table 19 – TrgOp

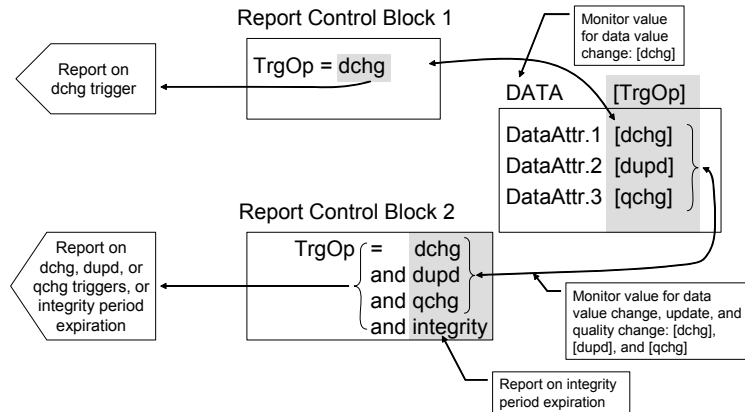
Kommentar [KS54]: See issue #338

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 .		

Kommentar [KS55]: #163

12 As depicted in Figure 13 the value of a **DataAttribute** that provides a specific **TrgOp** (trig-
13 ger option) shall be monitored for reporting and logging if the report control block has en-
14 abled the specific trigger option (**TrgOp**). In the upper example of Figure 13 the **TrgOp** is
15 **dchg**; the **TrgOp** of the **DataAttributes** is **dchg** for the first, **dupd** for the second, and **qchg**
16 for the last **DataAttribute**. Reports are sent on data changes only, because only **dchg** is

1 enabled in the report control block. In the second example, all changes will be reported. In
2 addition, a report will be sent on the expiration of the integrity period.



3
4 **Figure 13 – Relation of TrgOp and Reporting**

IEC 407/03

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

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

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

11 **10.2.2.4.4 Functional constrained data (FCD)**

12 The reference to an ordered collection of **DataAttributes** of a **DATA** having the same **func-**
13 **tional constraint (FC)** value shall be called **functional constrained data (FCD)**. The order
14 of the collection of the **FCD** shall be the order of the appearance of the **DataAttributes**
15 in the **DATA**.

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

19 **EXAMPLE 1** Figure 12 shows a **[MX] FCD** in the second line in the box of the bottom.

20 If the **DataAttributes** are elements of an array then the **FCD** shall be accompanied by the
21 **NumArrayElement** (value between 0 and m according to the instance of the array). The syn-
22 tax for a **FCD** of an array element shall be **FCD(NumArrayElement)**.

23 **EXAMPLE 2** The HWYE CDC in IEC 61850-7-3 uses the **ARRAY** type. The **[MX] FCD pbsAHar(3)** references the
24 values of **cVal**, **q** and **t** of the array element number 3 (in case the optional **MX** attributes like **instCVal** are not im-
25 plemented).

26 **10.2.2.4.5 Functional constrained data attribute (FCDA)**

27 A reference to a single **DataAttribute** or **DataAttributeComponent** of a **DATA** having a
28 specific **functional constraint (FC)** value shall be called **functional constrained data at-**
29 **tribute (FCDA)**.

30 **EXAMPLE 1** Figure 12 shows a **[MX] FCDA** in the fifth line in the box of the bottom.

Kommentar [KS56]: Re-
moved "(or attributes of a control block)" ... extra definition for
CBs

Kommentar [KS57]: Re-
moved "(or attributes of a control
block)" ... extra definition
fo CBs.

Kommentar [KS58]: #500
Last sentence in this paragraph
has been deleted because the
term FCD is already defined in
the first sentence.

Kommentar [KS59]: As a
consequence of adding **ARRAY**
type we have to define the use
of array elements in services.
The easiest way seems to be to
add the array element selection
to the FCD definition. The use
cases of FCD do not need to be
changed.

Kommentar [KS60]: Ac-
cording to editors' meeting in
Zug

Kommentar [KS61]: #500-
Last sentence in this paragraph
has been deleted because the
term FCD is already defined in
the first sentence.

1 If the **DataAttribute** or **DataAttributeComponent** is an element of an array then the
 2 **FCDA** shall be accompanied by the **NumArrayElement** (value between 0 and m according to
 3 the instance of the array). The syntax for a **FCDA** of an array element shall be
 4 **FCDA(NumArrayElement)**.

5 EXAMPLE 2 The HWYE CDC in IEC 61850-7-3 uses the ARRAY type. The [MX] FCDA phsAHar.cVal(3) references
 6 the value of cVal of the array element number 3.

7 NOTE A single measured value of a **DATA** (**FC = MX**) is referenced by an **FCDA**. The functional constrained data
 8 attribute is used, for example, to describe and to remotely create **DATA-SETS**. The syntax notation for **FCDA** is
 9 defined in a SCSM.

10 10.2.2.5 CompositeCDC [0..n]

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

12 10.2.2.6 SimpleCDC [0..n]

13 10.2.2.6.1 SimpleCDC syntax – General

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

15 10.2.2.6.2 COMMON-DATA class syntax

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

17 **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).		

18

19 **DataName**

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

Kommentar [KS62]: As a consequence of adding ARRAY type we have to define the use of array elements in services. The easiest way seems to be to add the array element selection to the FCDA definition. The use cases of FCDA do not need to be changed.

1 **DataRef – data ObjectReference**

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

3 The ObjectReference **DataRef** shall be:

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

4

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

6 **Presence**

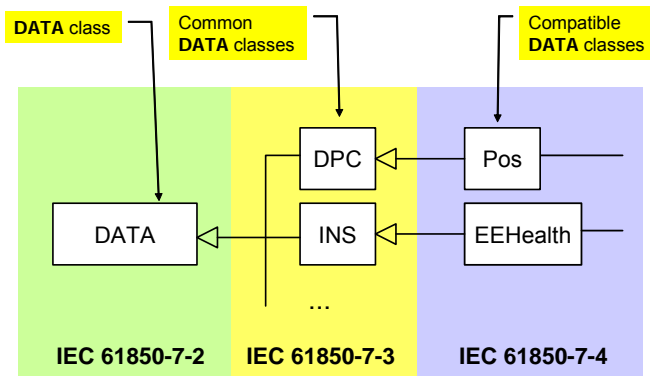
7 The attribute **Presence** of type **BOOLEAN** shall define if a **DATA** is mandatory (**Presence = TRUE**) or optional (**Presence = FALSE**).

9 **DataAttribute**

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

11 **10.3 Relation of DATA, common DATA, and compatible DATA classes**

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



15

16 **Figure 14 – Relation of DATA classes**

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

19 EXAMPLE The compatible **DATA** class with the name “**Pos**” represents a switch position. “**Pos**” is a specializa-
20 tion of the common **DATA** class “**DPC**” (double-point control). The **DATA** “**Pos**” may be used in one or several
21 **LOGICAL-NODES**.

22 **10.4 DATA class services**

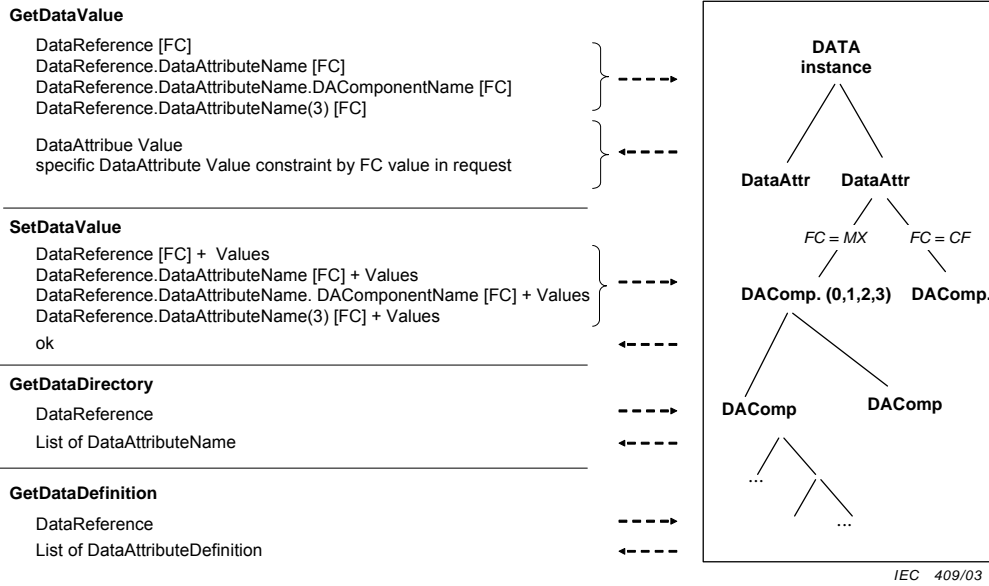
23 **10.4.1 General definitions and overview**

24 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

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



2
3

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

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

7 **10.4.2 GetDataValue**

8 **10.4.2.1 GetDataValue parameter table**

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

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

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

14

1 **10.4.2.2 Request**

2 **10.4.2.2.1 Reference**

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

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

7 **10.4.2.3 Response+**

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

10 **10.4.2.3.1 DataAttributeValue [1..n]**

11 The parameter **DataAttributeValue** [1..n] shall contain

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

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

15 **10.4.2.4 Response-**

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

18 **10.4.3 SetDataValue**

19 **10.4.3.1 SetDataValue parameter table**

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

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

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

25

26 **10.4.3.2 Request**

27 **10.4.3.2.1 Reference**

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

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

Kommentar [KS63]: Make the definition consistent with the rest.

Kommentar [KS64]: #195

1 10.4.3.2.2 DataAttributeValue [1..n]

- 2 The parameter **DataAttributeValue** [1..n] shall contain
- 3 – the values of all **DataAttributes** of a **DATA** referenced by **FCD**; or
- 4 – the value of a **DataAttribute** referenced by **FCDA**.

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

Kommentar [KS65]: Make the definition consistent with the rest

6 10.4.3.3 Response+

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

8 this parameter is SCSM specific.

Kommentar [KS66]: #411

9 NOTE 1 For the **SetDataValue** service, a successful result means that the service request was acceptable to the

10 server and that the server has attempted to move the value of each **DataAttribute** of the **DATA** requested by the

11 service to the corresponding application.

12 NOTE 2 The action to be taken by an application receiving the value for a **DATA** to be set is outside the scope of

13 this standard.

14 10.4.3.4 Response–

15 The parameter **Response–** shall indicate that the service request failed. The appropriate

16 **ServiceError** shall be returned.

17 The attempt to set a **DataAttribute** or an underlying component that is not available shall be

18 interpreted as a service failure.

Kommentar [KS67]: #386

19 10.4.4 GetDataDirectory

20 10.4.4.1 GetDataDirectory parameter table

21 A client shall use the **GetDataDirectory** service to retrieve the list of all **DataAttribu-**

22 **teNames** of the referenced **DATA** made visible and thus accessible to the requesting cli-

23 ent by the referenced **LOGICAL-NODE**.

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
DataReference
Response+
DataAttributeName [1..n]
Response–
ServiceError

26

27 10.4.4.2 Request

28 DataReference – data reference

29 The parameter **DataReference** shall contain the **ObjectReference** of a **DATA**. The **Ob-**

30 **jectReference** shall be **DataRef**.

1 **10.4.4.3 Response+**

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

4 **DataAttributeName [1..n]**

5 The parameter **DataAttributeName [1..n]** shall contain the names of all components below
6 the level the given by the **DataReference** of the **GetDataDirectory.request**.

7 **NOTE** The **DataAttributename** may be a **DataName** and or a **DataAttributeName**.

8 **EXAMPLE** A **GetDataDirectory.request** with (**DataReference** = **MMXU.PhsV**) would return **DataNames** (**phsA**,
9 **phsB**, **phsC**, **neut**, **net**, and **res**) and **DataAttributeNames** (**angRef**, **d**, **dU**, **cdcNs**). A **GetDataDirectory.request** with
10 (**DataReference** = **MMXU.PhsV.PhsA**) would return **DataAttributeNames** (**instCVal**, **cVal**, **range**, **rangeAng**, **q**, **t**, ...).

Kommentar [KS68]: According to editors' meeting in Zug.

11 **10.4.4.4 Response-**

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

14 **10.4.5 GetDataDefinition**

15 **10.4.5.1 GetDataDefinition parameter table**

16 A client shall use the **GetDataDefinition** service to retrieve the complete list of all **DataAt-**
17 **tribute** definitions of the referenced **DATA** made visible and thus accessible to the request-
18 ing client by the referenced **LOGICAL-NODE**.

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

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

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

Kommentar [KS69]: #415

23

24 **10.4.5.2 Request**

25 **DataReference – data ObjectReference**

26 The parameter **DataReference** shall contain the **ObjectReference** of the **DATA**. The **Ob-**
27 **jectReference** shall be **DataRef**.

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

29 **10.4.5.3 Response+**

30 **DataAttributeDefinition [0..n]**

Kommentar [KS70]: #415

31 The parameter **DataAttributeDefinition [0..n]** shall contain all **DataAttrName** and
32 **DataAttrType** of the first level and of all nested levels below of the referenced **DATA** and
33 the functional constraints of each **DataAttribute** where applicable.

Kommentar [KS71]: Make the definition consistent with the rest

1 **10.4.5.4 Response–**

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

4

5

1 **11 Modeling of control blocks**

2 **11.1 General**

3 Control blocks as defined for reporting (15.2), logging (15.3), GSE (16) and SV (17) are mod-
4 elled similar to DATA class. This clause defines the modelling approach for control blocks.

5 **Editors' Note** The control block modelling approach has been included to separate the Data from Control blocks. It
6 has been included after clause 10 (before the data set model) because the data set model could also refer to at-
7 tributes of a control block by FCCB.

8 **11.2 Control block models**

9 Control blocks shall be defined using the table notation as shown in Table 21.

10 **Table 21 – CB class definition**

CB class					
Attribute name	Attribute type	FC	TrgOp	r/w	Value/value range/explanation
CBName	ObjectName	-	-		Instance name of an instance of CB
CBRef	ObjectReference	-	-		Path-name of an instance of CB
Specific to report handler					
Attribute1	Type for Attribute1	FC	TrgOp	zz	
...	
Attributen	Type for Attributen	FC	TrgOp	zz	
cbNs	VISIBLESTRING255	EX			
Services Service1 Service2 ...					

11

12 **11.2.1 Attribute name**

13 The **Attribute name** shall be name of a control block attribute.

14 **11.2.1.1 CBName**

15 The attribute **CBName** shall be the name of the **CB** that unambiguously identifies the **CB**
16 within a **LOGICAL-NODE**.

17 **11.2.1.2 CBRef**

18 The attribute **CBRef** shall be the unique path-name of a **CB**.

19 The **ObjectReference CBRef** shall be:

LDName/LNName. CBName

20 **11.2.1.3 Attribute1 to Attributen**

21 Further attributes as required for a special control block.

1 **11.2.1.4 cbNs – control block name space**

2 This attribute shall be used to identify the name space of the control block.

3 **11.2.1.5 Services**

4 Services as required for a special control block.

5 **11.2.2 Attribute type**

6 The **Attribute type** shall be name of a control block attribute type.

7 **11.2.3 FunctionalConstraint for control blocks**

8 The **FunctionalConstraint (FC)** for control blocks shall be a property of the **control block**
 9 **attributes**.

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

13 **Table 22 – Functional constraint values**

FunctionalConstraint values			
FC	Semantic	Services allowed	Initial values/storage/ explanation
SP	Setpoint	DataAttribute shall represent a setting parameter information 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
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
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
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
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
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
MS	Multicast sampled value control	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
US	Unicast sampled value control	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
NOTE The possibility to write an Attribute may be further constrained by a view or an implementation.			

Kommentar [KS72]: #220; changes as proposed. "one whose value" removed

14

15 **11.2.4 TrgOp**

16 The definition shall be the same as defined in 10.2.2.4.3.

1 **11.3 Functional constrained for control block (FCCB)**

2 A reference of a single **Attribute** of a **control class** having a specific **functional con-**
3 **straint (FC)** value shall be called **functional constrained control block (FCCB)**.

4 EXAMPLE 1 [BR] BRCB1 referring to all control block attributes of BRCB1.

1 **12 DATA-SET class model**

2 **12.1 General**

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

Kommentar [KS73]: Added due to the split of data and control blocks.

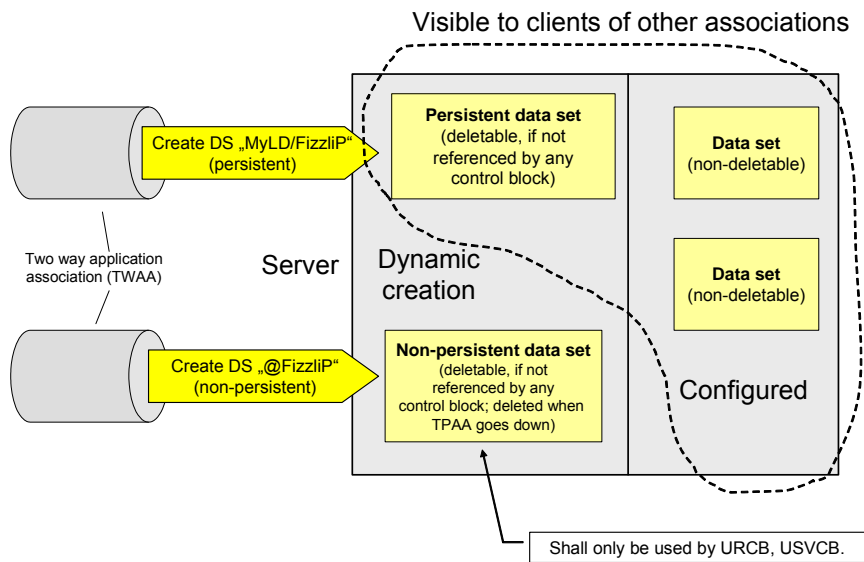
10 **NOTE 1** The membership and order of the **DATA** or **DataAttribute** in a **DATA-SET** can be retrieved with the
 11 **GetDataSetDirectory** service. The persistent existence of **DATA** and **DataAttribute** is expected as long as they
 12 are referenced as members of a **DATA-SET**. A system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

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

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

18 Any **DATA**, **DataAttributes**, or references to attributes of report control blocks in a
 19 **SERVER** may be referenced by one or more **DATA-SETS**.

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



26

IEC 410/03

27

Figure 16 – Dynamic creation of data set instances

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

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

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

13 **12.2 DATA-SET class definition**

14 **12.2.1 DATA-SET class syntax**

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

16 **Table 23 – 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		

17
 18

19 **12.2.2 DATA-SET class attributes**

20 **12.2.2.1 DSName**

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

23 **12.2.2.2 DSRef**

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

25 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

26

1 **12.2.2.3 DSMemberRef [1..n] – data set member reference**

2 The attribute **DSMemberRef** shall define the functional constrained data (**FCD**) or functional
 3 constrained data attribute (**FCDA**) of **DATA** or a reference to an attribute of a report control
 4 block (**FCCB**).

Kommentar [KS74]: Due to the split of Data and CBs

5 An IED which claims to support dynamic creation of datasets (**CreateDataSet**) shall be able
 6 to receive (as a server), and process (as a server) any valid **FCD**, **FCDA** or **FCCB** definition
 7 contained in the **CreateDataSet** request. An IED which claims to support configuration of
 8 datasets (via **SCL**) shall be able to process (as a server or as a client) any valid **FCD**, **FCDA**
 9 or **FCCB** definition contained in the corresponding **SCL** file.

Kommentar [KS75]: Removed "send (as a client)," due to editors' meeting in Zug

Kommentar [KS76]: Removed "or as a client" due to editors' meeting in Zug.

10 The value of a member of a **DATA-SET** retrieved, set, reported, or logged shall be deter-
 11 mined by the functional constrained **DATA** (**FCD**), functional constrained data attribute
 12 (**FCDA**) or functional constrained control block (**FCCB**).

Kommentar [KS77]: Added due to Tissue # 426

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

16 **12.3 DATA-SET class services**

17 **12.3.1 Overview**

18 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

19
 20 **12.3.2 GetDataSetValues**

21 **12.3.2.1 GetDataSetValues parameter table**

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

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

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

27

1 **12.3.2.2 Request**

2 **DataSetReference – data set ObjectReference**

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

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

7 **12.3.2.3 Response+**

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

9 The parameter **DataAttributeValue** shall contain values of a member of the **DATA-SET**.
10 The value of the **DataAttributes** of the **DATA** may be simple or complex depending on the
11 definition of the **DATA**. For complex **DataAttrTypes** the values of all **DataAttributes** of all
12 nesting levels shall be returned.

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

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

Kommentar [KS78]: #503

16 **12.3.2.4 Response–**

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

19 **12.3.3 SetDataSetValues**

20 **12.3.3.1 SetDataSetValues parameter table**

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

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

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

25

26 **12.3.3.2 Request**

27 **12.3.3.2.1 DataSetReference – data set ObjectReference**

28 The parameter **DataSetReference** shall define the **ObjectReference** of a **DATA-SET**. The
29 **ObjectReference DataSetReference** shall be one of the following two options:

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

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

2 12.3.3.2 DataAttributeValue [1..n]

3 The parameter **DataAttributeValue** shall contain a value of a member of the **DATA-Set**.
 4 The value of the **DataAttribute** of the **DATA** may be simple or complex depending on the
 5 definition of the **DATA**. For complex **DataAttrTypes** the values of all **DataAttributes** of all
 6 nesting levels shall be contained.

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

Kommentar [KS79]: #503

8 12.3.3.3 Response+

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

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

12 A successful result shall return the following parameter.

13 Result

14 The parameter **Result** shall return a list, specified in the order of the **ObjectReferences** of
 15 the **DATA** that are referenced in the **DATA-SET**. This list shall indicate, for each **DATA**, ei-
 16 ther a confirmation that the service **SetDataSetValue** to the referenced instance succeeded
 17 or a reason why the service **SetDataSetValue** to the referenced **DATA** failed.

18 12.3.3.4 Response–

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

21 12.3.4 CreateDataSet

22 12.3.4.1 CreateDataSet parameter table

23 The client shall use the **CreateDataSet** service to request the server to create a **DATA-SET**
 24 with a list of members defined with the functional constrained data (**FCD**), functional con-
 25 strained data attribute (**FCDA**) or a reference to an attribute of a report control block (**FCCB**)
 26 made visible and thus accessible to the requesting client.

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

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

29

1 **12.3.4.2 Request**

2 **12.3.4.2.1 DataSetReference – data set ObjectReference**

3 The parameter **DataSetReference** shall define the **ObjectReference** of **DATA-SET** that is to
4 be created. The **ObjectReference DataSetReference** shall be one of the following two op-
5 tions:

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

8 **12.3.4.2.2 DSMemberRef [1..n] – data set member ObjectReference**

9 The parameter **DSMemberRef [1..n]** shall define all functional constrained data (**FCD**),
10 functional constrained data attributes (**FCDA**) of a **DATA** or references to the attributes of a
11 report control block (**FCCB**).

Kommentar [KS80]: Adde due to the split of Data and CB.

12 **12.3.4.3 Response+**

13 The parameter **Response+** shall indicate that the service request succeeded. If one of the
14 members to be defined is not available to that client then the service shall fail.

Kommentar [KS81]: Make definition consistent

15 **12.3.4.4 Response–**

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

18 **12.3.5 DeleteDataSet**

19 **12.3.5.1 DeleteDataSet parameter table**

20 The client shall use the **DeleteDataSet** service to request the server to delete a **DATA-SET**
21 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
Response+
Response–
ServiceError

24

25 **12.3.5.2 Request**

26 **12.3.5.2.1 DataSetReference – data set ObjectReference**

27 The parameter **DataSetReference** shall define the **ObjectReference** of a **DATA-SET** that
28 shall be deleted. The **ObjectReference DataSetReference** shall be one of the following
29 two options.

- 30 – **LDName/LNName.DataSetName** to delete a dynamically created **persistent DATA-**
31 **SET**, or
- 32 – **@DataSetName** to delete a **non-persistent DATA-SET**.

1 **12.3.5.3 Response+**

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

3 **12.3.5.4 Response–**

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

6 **12.3.6 GetDataSetDirectory**

7 **12.3.6.1 GetDataSetDirectory parameter table**

8 The client shall use the **GetDataSetDirectory** service to retrieve the list of the **Object-**
9 **References** of all data set members referenced by the **DATA-SET** made visible and thus ac-
10 ccessible to the requesting client.

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

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

13

14 **12.3.6.2 Request**

15 **DataSetReference – data set ObjectReference**

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

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

20 **12.3.6.3 Response+**

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

23 **DSMemberRef [1..n] – data set member ObjectReference**

24 The parameter **DSMemberRef** shall contain the **ObjectReferences** of the members of the
25 **DATA-SET**.

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

27 **12.3.6.4 Response–**

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

1 **13 Substitution model**

2 The substitution model provides the substitution of values of **DataAttributes** whose func-
3 tional constraint equals **MX** (for analogue values) or **ST** (for status values). Basically, substi-
4 tution applies to **DataAttributes** with FC (= **MX** or **ST**) and to the associated quality attrib-
5 ute. When substitution is enabled for a specific **DataAttribute**, the **DATA** shall provide the
6 substituted values instead of the process value to the clients.

Kommentar [KS82]: #513

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

Kommentar [KS83]: The service parameter has been removed from the service. Now it is an **DataAttribute** of a controllable data object. Need to revise this statement.

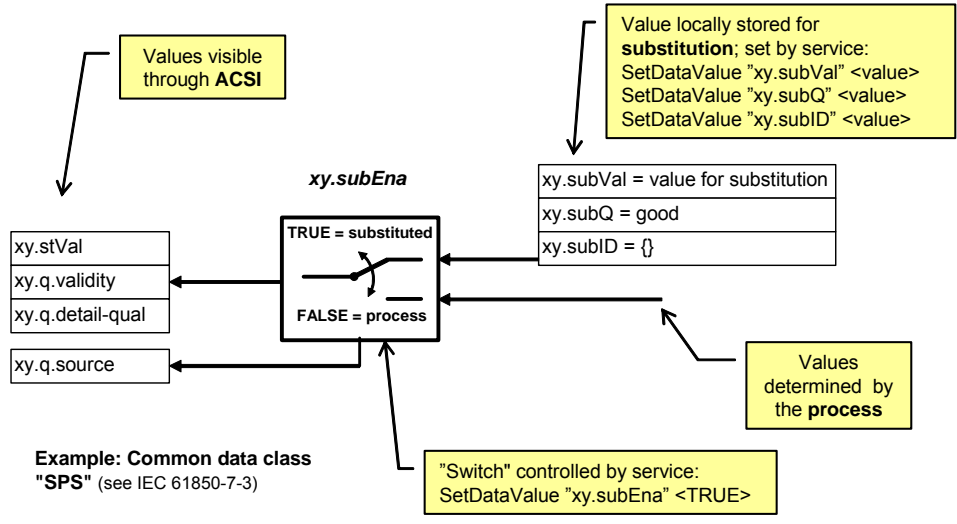
10 In the typical use case for substitution, an operator on the client side enters manually a value
11 for a **DataAttribute** located in a specific device. The client sets the **DataAttribute** to the
12 value entered. If a client accesses the value of that **DataAttribute** (for example, using a
13 **GetDataValue** service or subscribing to a report) the client shall receive the manual entered
14 (substituted) value instead of the value determined by the process.

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

- 16 – **subEna** (enable substitution): The current process value shall be replaced by the value
17 provided in the **DataAttribute subVal**.
- 18 – **subVal**, **subMag**, and **subCMag** (values for substitution of process values): The current
19 process value shall be replaced by the value provided by the **DataAttribute subVal**,
20 **subMag** and **subCMag** respectively.
- 21 – **subQ** (value for substitution of quality): The current process value shall be replaced by
22 the value provided by the **DataAttribute subQ**.
- 23 – **subID** (value to indicate the initiator of the substitution).

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

26 The concept of substitution is shown in Figure 17. Usually, input from the process or the result
27 of the calculation from a function provides the value of a **DataAttribute** (in that case,
28 the source is called "process"). In case of substitution, the value of a **DataAttribute** may be
29 provided by an operator making use of a client. This selection of the source of the value
30 (substitution value or process value) shall be controlled by the service **SetDataValue**
31 ("**xy.subEna**" <TRUE>) to substitute or **SetDataValue** ("**xy.subEna**" <FALSE>) to unsubsti-
32 tute. The service **SetDataValue** ("**xy.subVal**" <value for substitution>) shall be used to set
33 the substituted value. There may be cases, where a local automatic function disables substi-
34 tution, for example, if blocking of information exchange is disabled or communication is no
35 longer interrupted.



IEC 411/03

Figure 17 – Principles of substitution

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

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

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

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

Kommentar [KS84]: According to Tissue # 374

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

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

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

1 **14 SETTING-GROUP-CONTROL-BLOCK class model**

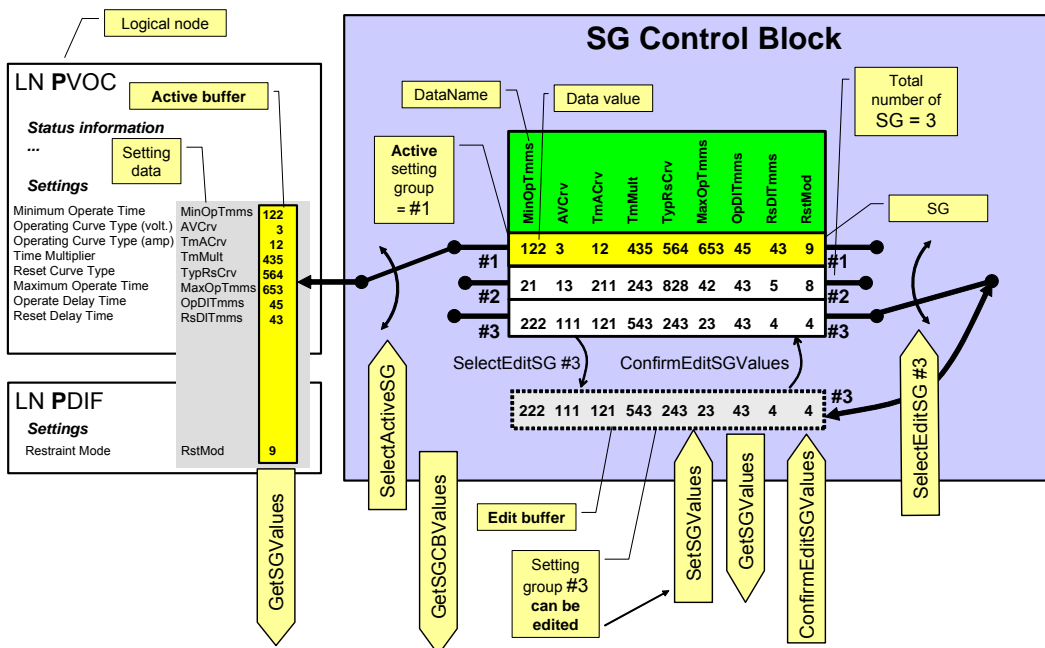
2 **14.1 General**

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**.
 6 Values 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**”
 13 (voltage controlled/dependent time overcurrent according to IEC 61850-7-4) comprises eight
 14 **DATA** for settings (**LN PDIF** has one **DATA** for settings) – (**MinOpTmms**, ..., **RstrMode**).
 15 The **SGCB** “**SG Control**” provides three **SGs** (#1, #2, and #3) each with independent values
 16 for the nine **DATA**. Each **SG** contains nine values (one for each of: **MinOpTmms**, ...,
 17 **RstrMode**). The members of the active **SG** are referenced by the **ObjectReferences** of the
 18 **DATA** with functional constraint **SG**. The members of the **SG** in the “edit buffer” are refer-
 19 enced by the **ObjectReferences** of the **DATA** with functional constraint **SE**.



IEC 412/03

20
21

22

Figure 18 – Basic model of the settings model

Kommentar [KS85]: #536, data name changed in figure; min -> max

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**
 3 determines which values (of **SG #1, #2, or #3**) shall be copied to the “active buffer” and be
 4 used 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 14.2 SGCB class definition

15 14.2.1 SGCB class syntax

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

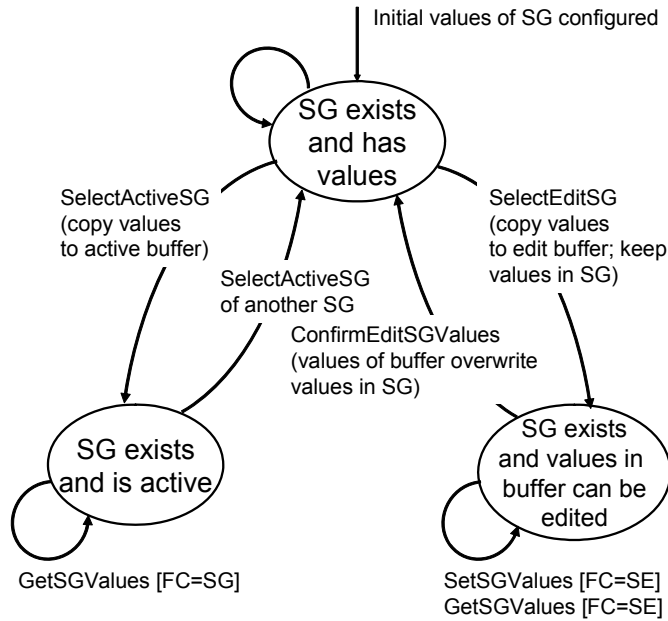
17 Clients should use the existence of a **SGCB** to determine if the **LOGICAL-DEVICE** contains **SGs**.

18 **Table 24 – 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 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

Kommentar [KS86]: Text in figure revised according to official comment JP4

1

2

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

5 **14.2.2 SGCB class attributes**

6 **14.2.2.1 SGCBName – setting group control name**

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

8 **14.2.2.2 SGCBRef – setting group control ObjectReference**

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

10 The ObjectReference **SGCBRef** shall be:

LDName/LLNO.SGCB

11

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

13 **14.2.2.3 NumOfSG – number of setting groups**

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

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

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

1 14.2.2.4 ActSG – active setting group

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

6 14.2.2.5 EditSG – edit setting group

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

Kommentar [KS87]: Text revised according to official comment JP4

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

15 14.2.2.6 CnfEdit – confirm editing

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

17 14.2.2.7 LActTm – last activation time

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

20 14.3 SGCB class services

21 14.3.1 Overview

22 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

23

24 14.3.2 SelectActiveSG

25 14.3.2.1 SelectActiveSG parameter table

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

27

Parameter name
Request
SGCBReference
SettingGroupNumber
Response+

Response-
ServiceError

1 **14.3.2.2 Request**

2 **14.3.2.2.1 SGCBReference**

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

4 **14.3.2.2.2 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 **14.3.2.3 Response+**

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

13 **14.3.2.4 Response-**

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

16 **14.3.3 SelectEditSG**

17 **14.3.3.1 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
21 (confirming) the setting group in the edit buffer after an association was down. After loss of
22 an association the **SelectEditSG** service shall be re-issued to copy the values of the se-
23 lected **SG** to 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 **14.3.3.2 Request**

28 **14.3.3.2.1 SGCBReference**

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

1 The **ObjectReference** **SGCBReference** shall be:

2 **LDName/LLNO.SGCB**

3 **14.3.3.2 SettingGroupNumber**

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

7 **14.3.3.3 Response+**

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

9 **14.3.3.4 Response–**

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

12 **14.3.4 SetSGValue**

13 **14.3.4.1 SetSGValue parameter table**

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

17 Setting new value shall become effective only after the client has confirmed the value by is-
18 suing the service **ConfirmEditSGValue**.

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
Reference
DataAttributeValue [1..n]
Response+
Response–
ServiceError

21

22 **14.3.4.2 Request**

23 **14.3.4.2.1 Reference**

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

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

28 **14.3.4.2.2 DataAttributeValue [1..n]**

29 The parameter **DataAttributeValue** shall contain

30 – the values of all **DataAttributes** of a **DATA** referenced by **FCD**; or

1 – the value of a **DataAttribute** referenced by **FCDA**
 2 of the **SG** identified by the value of the attribute **EditSG** of the **SGCB**.

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

4 **14.3.4.3 Response+**

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

6 **14.3.4.4 Response-**

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

9 **14.3.5 ConfirmEditSGValue**

10 **14.3.5.1 ConfirmEditSGValue parameter table**

11 A client shall use the **ConfirmEditSGValue** service to confirm that the value of the **SG** (iden-
 12 tified by the attribute **EditSG**) set with the service **SetSGValue** shall overwrite the old value
 13 of the **SG** of the **SGCB** made visible and thus accessible to the requesting client by the refer-
 14 enced **LLNO**.

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
SGCBReference
Response+
Result
Response-
ServiceError

17

18 **14.3.5.2 Request**

19 **SGCBReference**

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

21 **14.3.5.3 Response+**

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

23 **14.3.5.4 Response-**

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

26 **14.3.6 GetSGValue**

27 **14.3.6.1 GetSGValue parameter table**

28 A client shall use the **GetSGValue** service to retrieve the value of **DATA** of **SGs** made visible
 29 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
Reference
Response+
DataAttributeValue [1..n]
Response–
ServiceError

3

4 14.3.6.2 Request

5 Reference

6 The parameter **Reference** shall define the **functional constrained data (FCD)** or **func-**
7 **functional constrained data attributes (FCDA)** of the **DATA** whose **DataAttribute** value are
8 to 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 14.3.6.3 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 14.3.6.4 Response–

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

22 14.3.7 GetSGCBValue

23 14.3.7.1 GetSGCBValue parameter table

24 A client shall use the **GetSGCBValue** service to retrieve the list of attribute values of the ref-
25 erenced **SGCB** made visible and thus accessible to the requesting client by the referenced
26 **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 **14.3.7.2 Request**

5 **14.3.7.2.1 SGCBReference**

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

7 **14.3.7.2.2 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
10 the **LLNO**. The value shall be **SP**.

11 **14.3.7.3 Response+**

12 **14.3.7.3.1 NumberOfSettingGroup – number of setting group controls**

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

15 **14.3.7.3.2 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 **14.3.7.3.3 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 **14.3.7.3.4 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 **14.3.7.4 Response-**

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

27

1 15 REPORT-CONTROL-BLOCK and LOG-CONTROL-BLOCK class models

2 15.1 Overview

3 Reporting and logging meets a number of crucial requirements for event-driven information
4 exchange. The data transfer models described in this clause provide mechanisms for trans-
5 ferring data values caused by well-defined conditions from a logical node to one client or
6 storing the 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
9 is 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
12 keeping 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.

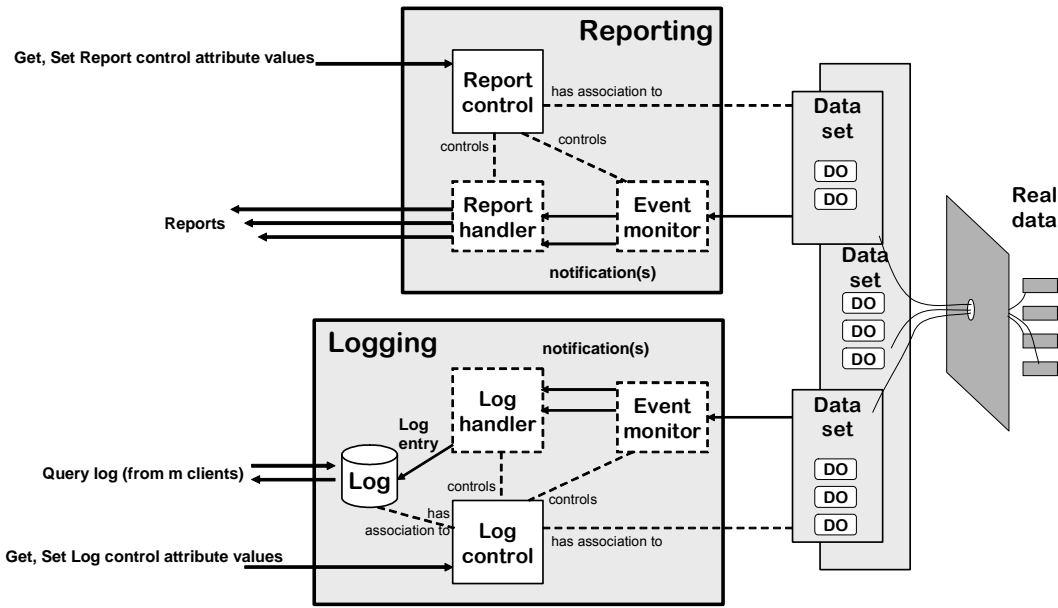
20 Reporting and logging as well as the basic services of the data model provide flexible data
21 retrieval schemas, for example:

- 22 – change-of-state notification of clients: immediate reports,
- 23 – sequence-of-events: keeping reports in sequence or storing and querying sequences of
24 log 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-
28 oriented (**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

Kommentar [hf88]: Changed event(s) to notification

4

5 The reporting model is composed of three building blocks. The logging model has four build-
6 ing 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
10 the state of the real data and the attributes of the control class, when to generate a notific-
11 ation to the appropriate handlers (e.g. Log or Report Handler). This notification includes the
12 data values 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 num-
15 ber 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 notific-
18 ations. 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
22 which the report handler received the first notification that is used to form an EntryID. The re-
23 port handler decides when and how to send a report to the subscribed client. The log handler
24 stores a log entry to the log.

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

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

3 **15.2 REPORT-CONTROL-BLOCK class model**

4 **15.2.1 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
9 client exclusively shall “own” that instance and shall receive reports from that instance of re-
10 port 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
13 options data-change, quality-change, and data-update) issue immediate sending of re-
14 ports or buffer the events (to some practical limit) for transmission, such that values of
15 **DATA** are not lost due to transport flow control constraints or loss of connection. **BRCB**
16 provides the 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 re-
19 ports on a “best efforts” basis. If no association exists, or if the transport data flow is not
20 fast 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
24 a Logical-Node. The number of instances, that may be visible to a specific client, is a local
25 implementation 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 **15.2.2 BUFFERED-REPORT-CONTROL-BLOCK (BRCB) class definition**

32 **15.2.2.1 BRCB class Syntax**

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

1

Table 25 – 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.					

Kommentar [hsf89]: How are we going to note that this is an edition 2 BRCB?

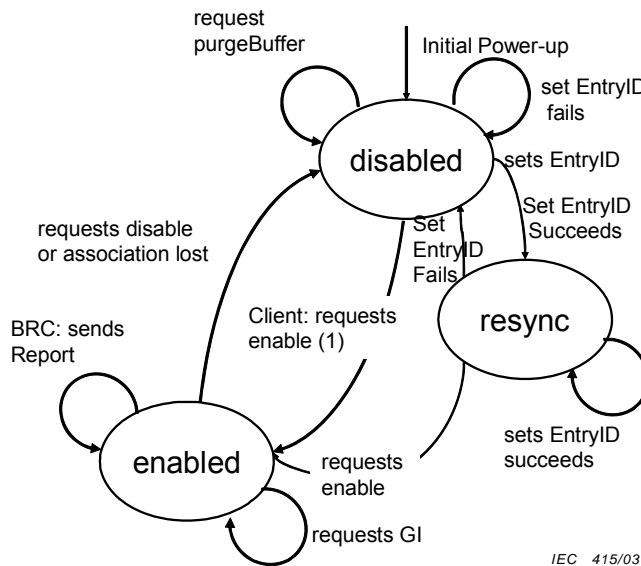
Kommentar [hsf90]: Tissue 322

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
 3 of performing this configuration is a local issue. However, it is assumed that one or more co-
 4 operating clients (e.g. primary and secondary SCADA systems) may be allowed to have ac-
 5 cess to a single **BRCB**. For each set of cooperating clients, for which the server is to provide
 6 buffered reporting service, different **BRCBs** should be provided.

7



8

9

Figure 21 – BRCB state machine

Kommentar [hsf91]: Changed diagram

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.

Kommentar [hf92]: Changed on 2/07

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

Kommentar [hf93]: Added 2/07

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

4 If the value of the set EntryID does not exist within the queue of entries, a ServiceError
5 of parameter-value-inappropriate shall be returned and the BRCB state shall transition
6 to **disabled**.

7 If the EntryID value, in the SetBRCBValues, is zero(0), this value is reserved to be
8 used by the client to resync to the first entry in the queue, the state shall transition from
9 resync 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
12 **enabled**.

Kommentar [KS94]: #495

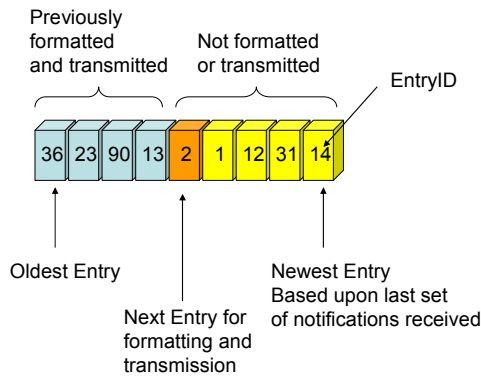
13 **enabled:** the **BRCB** shall generate reports for the buffered events and new events as speci-
14 fied in the **BRCB**.

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
22 a pointer to the next entry to be queue for formatting and transmission.



23

24 **Figure 22 – General Queue of Entries for Report Handler**

25 **15.2.2.2 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 **15.2.2.3 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 15.2.2.4 RptID – report identifier

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

Kommentar [hf95]: Changed 2/07

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 15.2.2.5 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**.

Kommentar [KS96]: #495

Kommentar [hf97]: Changed 2/07

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 ex-
19 cept disabling and activating general-integration.

20 15.2.2.6 DataSet – 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 **SetReportControl-
27 Values** service shall indicate success. If the **DataSet** reference is invalid, then the **SetRe-
28 portControlValues** service shall indicate failure of **badValue** and the value of the attribute
29 shall remain **unchanged**.

Kommentar [hf98]: Added 2/07

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

33 15.2.2.7 ConfRev – configuration revision

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

37 – any deletion of a member of the **DATA-SET**;

- 1 - the reordering of members of the **DATA-SET**; and
- 2 - Successful SetxRCBValues of the DatSet attribute where the DatSet attribute value
- 3 changes.
- 4 The counter shall be incremented when the configuration changes. At configuration time, the
- 5 configuration tool will be responsible for incrementing/maintaining the ConfRev value. When
- 6 configuration changes occur due to SetxRCBValues, the IED shall be responsible for incre-
- 7 menting the value of ConfRev.

8 The initial value for ConfRev is outside the scope of this part of IEC 61850. The value of 0

9 shall be reserved. The value of ConfRev, upon a restart of the IED, is a local issue.

Kommentar [hsf99]: Further clarified.

10 Note: There are two possible options for the initializing of the value of ConfRev and the associated configuration:

11 the value is restored to the original value of the base local configuration or the value is retained from the the con-

12 figuration prior to restart. The PICS shall clarify which is implemented.

13 **15.2.2.8 OptFlds – optional fields to include in report**

14 The attribute **OptFlds** shall be the client-specified optional fields to be included in the report

15 issued by this **BRCB**. This attribute defines a subset of the optional header fields of the report

16 (see 14.2.3.2.2.1) that shall be included in the report:

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

26 If a **BRCB** does not support one of the above options, then an attempt to set the correspond-

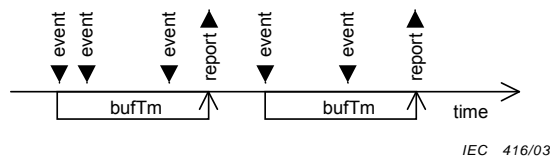
27 ing bit to TRUE shall cause a negative response of the **SetBRCBValues** service.

28 **15.2.2.9 BufTm – buffer time**

29 The attribute **BufTm** (see Figure 23) shall specify the time interval in milliseconds for the

30 buffering of internal notifications caused by data-change (**dchg**), quality-change (**qchg**),

31 data-update (**dupd**) by the **BRCB** for inclusion into a single report.



32 **Figure 23 – Buffer time**

IEC 416/03

34 Upon receipt of the first set of internal notification of events of the referenced **DATA-SET**, the

35 **BRCB** shall start a timer of the duration buffer time. The number of events within the internal

36 notification is a local issue. When the timer expires, the **BRCB** shall combine all internal noti-

37 fications that have been received during the time interval into a single report. The next inter-

38 nal notification following the timer expiration shall signal the new start of that timer.

Kommentar [hsf100]: Changed text to allow multiple events to be notified within a single internal notification. Tissue 305.

1 The default value of 0 shall be reserved to indicate that the buffer time attribute is not to be
 2 used by the **BRCB**. In this case, each internal notification shall cause the **BRCB** to send a
 3 single report. The value shall be settable in 1 ms increments and shall be able to convey up
 4 to 1 h of buffer time.

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

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

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

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

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

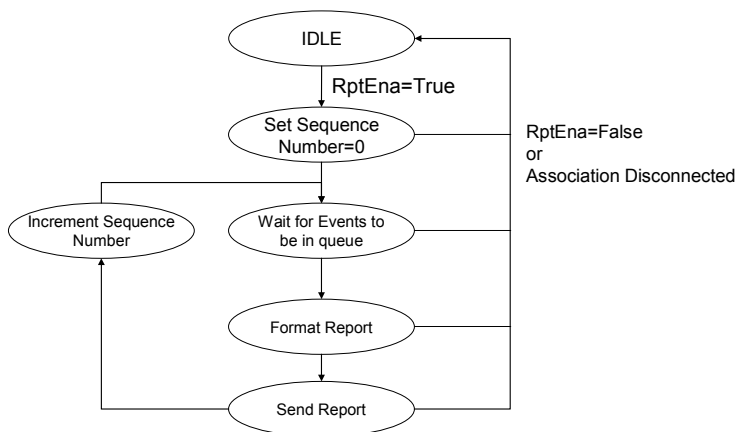
22 A **SetReportControlValues** of the attribute **BufTm** shall have the same effect as setting **pur-**
 23 **geBuf** to **TRUE** if the attribute value changes.

Kommentar [hsf101]: Tissue 322

24 **15.2.2.10 SqNum – sequence number**

25 The attribute **SqNum** shall specify the sequence number for each **BRCB** that has report en-
 26 able set to **TRUE**. This number is to be incremented by the **BRCB** for each report generated
 27 and sent. The increment shall occur once the **BRCB** has formatted the report and and re-
 28 quested for transmission.,

Kommentar [hsf102]: Added state machine and text to clarify the generation of the Sequence Number. Tissue 297



29
 30 **Figure 24 – State Machine for Sequence Number Generation**
 31

1 Figure 24 depicts the logical state machine used in the generation of SqNum. The transition
2 from the IDLE state is caused by the transition of RptEna changing from a value of FALSE to
3 a value of TRUE. Upon that transition, the value of SqNum shall be set to zero(0). As Reports
4 are sent, the value of SqNum shall be incremented. When the maximum value of SqNum is
5 obtained, the next value shall be a value of zero(0). .

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

8 **15.2.2.11 TrgOps – trigger options**

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

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

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

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

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

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

Kommentar [hsf103]: Tissue 322

27 **15.2.2.12 IntgPd – integrity period**

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

Kommentar [hsf104]: Changed text to reflect that integrity may be set in addition to other trigger options.

32 If a **BRCB** does not support integrity period then an attempt to set the **IntgPd** attribute to a
33 value greater than 0 shall cause a negative response of the **SetReportControlValues** ser-
34 vice.

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

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

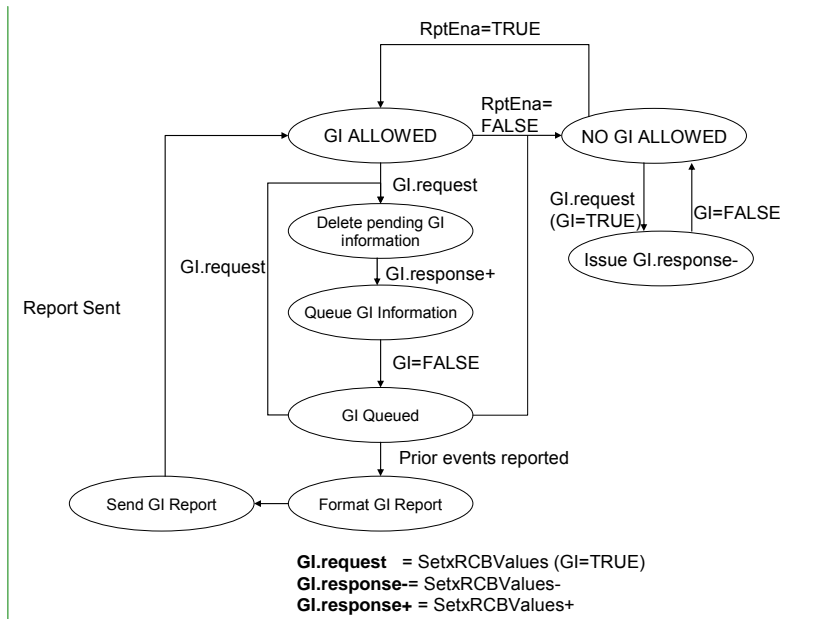
Kommentar [hsf105]: Tissue 322

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

1 **15.2.2.13 GI – general-interrogation**

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

Kommentar [hsf106]: Clarification of how GI is supposed to work



6 **Figure 25 – Logical State Machine for General Interrogation**

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

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

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

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

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

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

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

- 1 Note While GI=TRUE, subsequent GI.requests shall not result in additional general-interrogation processing.
- 2 If a client attempts to set GI=FALSE a positive response shall be issued. However, GI pend-
- 3 ing processing will not be impacted.
- 4 If a **BRCB** does not support **general-interrogation** then an attempt to set the **GI** attribute to
- 5 TRUE shall cause a negative response of the **SetReportControlValues** service containing
- 6 not-supported.
- 7 If a BRCB contains TrgOps.GI = FALSE then an attempt to set the GI attribute to TRUE shall
- 8 cause a SetReportControlValues positive response. No GI report shall be generated.

9 **15.2.2.14 PurgeBuf – purge buffer**

10 The attribute **PurgeBuf** shall indicate the request to discard buffered events. After setting to
11 TRUE, the **BRCB** shall discard all buffered events that have not yet been sent to the client.
12 After discarding the buffered events, this attribute shall be automatically set to FALSE by
13 the **BRCB**.

14 **15.2.2.15 EntryID – entry identifier**

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

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

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

- 22 – When the BRCB state is RptEna=FALSE: a GetBRCBValues shall return the EntryID
- 23 value that represents the last (i.e..newest) entry that has been entered into the buffer.
- 24 – When the BRCB RptEna=TRUE: The value of EntryID, returned in a GetBRCBValues re-
- 25 sponse, shall be the EntryID of the last EntryID formatted and queued for transmission.

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

28
29 NOTE 1 Clients needing to resync to the first set of entries in the buffer should use SetBRCBValues with an En-
30 tryID whose value is all zeros(0)

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

35 **15.2.2.16 TimeOfEntry – time of entry**

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

39 The value, returned in a GetBRCBValues response, shall provide the time stamp of the En-
40 tryID whose value is exposed in the control block. The value exposed for TimeOfEntry, when
41 the value of EntryID is zero(0), is a local issue.

Kommentar [hf107]: Chan-
ged 2/07

Kommentar [hsf108]: En-
tryID

1 **15.2.2.17 ResvTms – reservation time**

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

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

21 **15.2.3 BRCB class services**

22 **15.2.3.1 Overview**

23 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

24

25 **15.2.3.2 Report**

26 **15.2.3.2.1 Report parameter table**

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

Parameter name
Request
ReportFormat

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

1 **15.2.3.2.2 Request**

2 **15.2.3.2.2.1 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 26 – 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 15.2.3.2.2.10.
^a The type and value of this parameter shall be derived from the respective attribute of the BRCB or URCB .		

Kommentar [hsf109]: Changed from TriggerOptions to ReasonForInclusion

Kommentar [hsf110]: tisque 339

Kommentar [KS111]: #497

6

1 15.2.3.2.2.2 RptID – report ID

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

3 15.2.3.2.2.3 OptFlds – optional fields to include in report

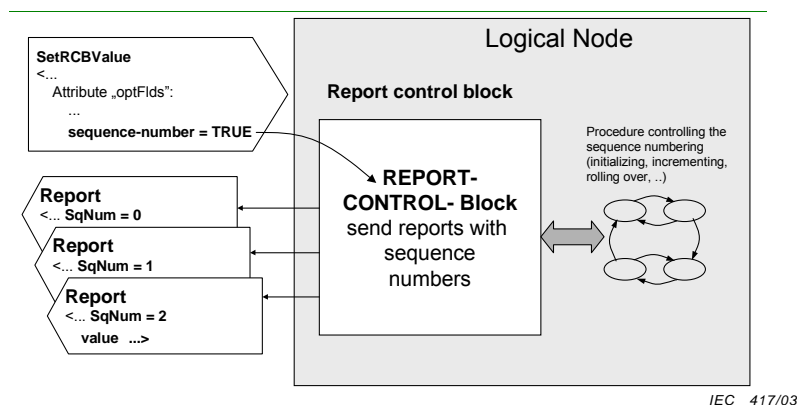
4 The parameter **OptFlds** shall specify which of the optional fields (**sequence-number**, **report-time-stamp**, **reason-for-inclusion**, **data-set-name**, **data-reference**, **buffer-overflow**, **entryID**, or **conf-revision**) shall be included in the Report.

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

8 15.2.3.2.2.4 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 se-
13 quence 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
16 be 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 15.2.3.2.2.5 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 in-
24 cremented for each subreport generated and sent based upon the report control instance.
25 The increment shall occur once the server has formatted the subreports and queued the sub-
26 report to the next lower protocol layer. The first subreport of the report shall have a subse-
27 quence number of zero. The subsequence number shall roll over to 0 after all subreports of
28 one specific 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
31 shall 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 **15.2.3.2.2.6 MoreSegmentsFollow – more report segments follow**

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

7 **15.2.3.2.2.7 DataSet – data set reference**

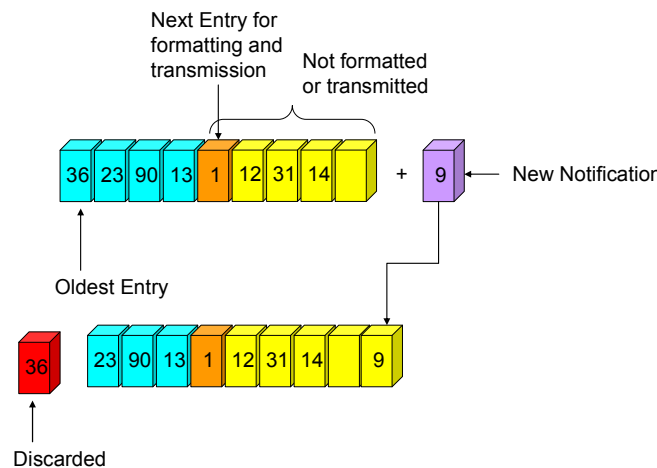
8 The parameter **DataSet** shall be derived from the respective attribute in the **BRCB**.

9 **15.2.3.2.2.8 BufOvfl – Possible Information Loss**

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

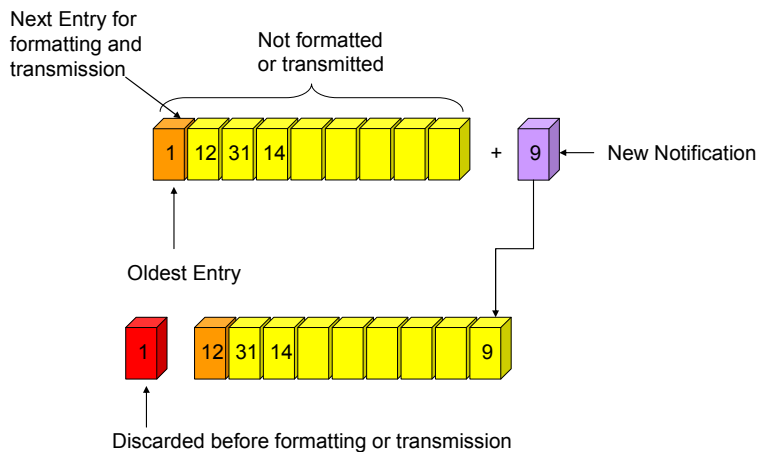
13 The **Report Handler** shall set **BufOvfl= TRUE** in the first report that is sent after the transi-
 14 tion from disabled to enabled. Subsequent reports shall have **BufOvfl = FALSE**.

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.). Implemen-
 17 tations shall discard the oldest Entry(s) in the queue in order to accept new notifications. If
 18 one of Entries discarded causes the Report Handler to move the pointer to the Next Entry for
 19 transmission, the implementation shall indicated **BufOvfl=TRUE** in the next entry that is for-
 20 matted 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** 3 **State**

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

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

9 **15.2.3.2.2.9 Entry**

10 **TimeOfEntry – report time stamp**

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

14 NOTE The event "time at which the report was generated" is determined by a specific implementation.

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

18 Reports with the same sequence number but different subsequence numbers shall use the
19 same **TimeOfEntry**.

20 **EntryID – entry identifier**

21 For the definition see 15.2.2.15.

22 **EntryData [1..n]**

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

1 **DataRef**

2 The parameter **DataRef** shall contain the reference of the DataSet member that is be-
3 ing reported.

4 **Value**

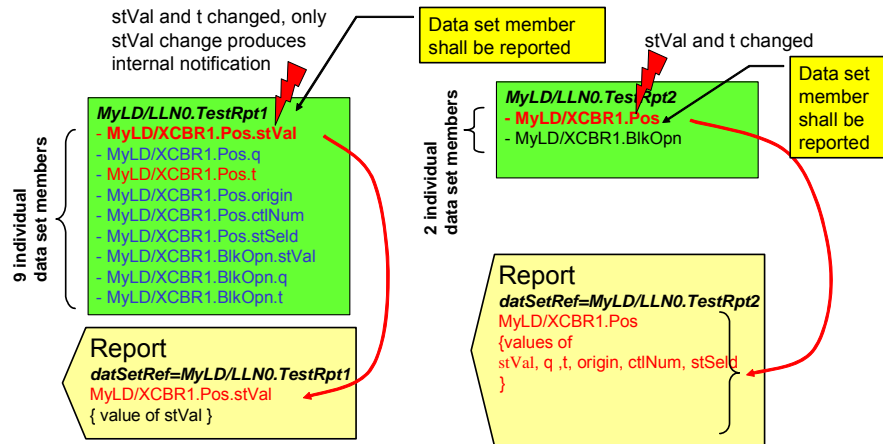
5 The parameter **Value** shall contain the **DataAttribute** values included in the report.

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

9 **BufTm = 0**

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

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



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

IEC 418/03

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

23 **BufTm > 0**

24 In the case of (**BufTm > 0**) the values of all members of a **DATA-SET** containing at-
25 tributes that have changed with respect to TrgOps during the bufferTime interval shall
26 be included that produced internal notifications during the buffer time. Further con-
27 straints apply; see 14.2.2.9 for additional details on **BufTm**.

28 **ReasonCode – reason for inclusion**

29 The reason for inclusion shall be included in the report if the optional fields to include in
30 report attribute (**OptFids**) of the **BRCB** includes reason for inclusion (=TRUE); other-
31 wise, it shall be omitted. The value for the reason for inclusion shall be set according to

1 the **TrgOps** that caused the creation of the report. The value range for reasons for in-
2 clusion shall be as defined in 15.2.3.2.2.10.

3 15.2.3.2.2.10 ReasonCode (ReasonForInclusion)

4 The values conveyed by ReasonForInclusion shall be a PACKEDLIST as defined in Table 27.

5 **Table 27 – 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.			

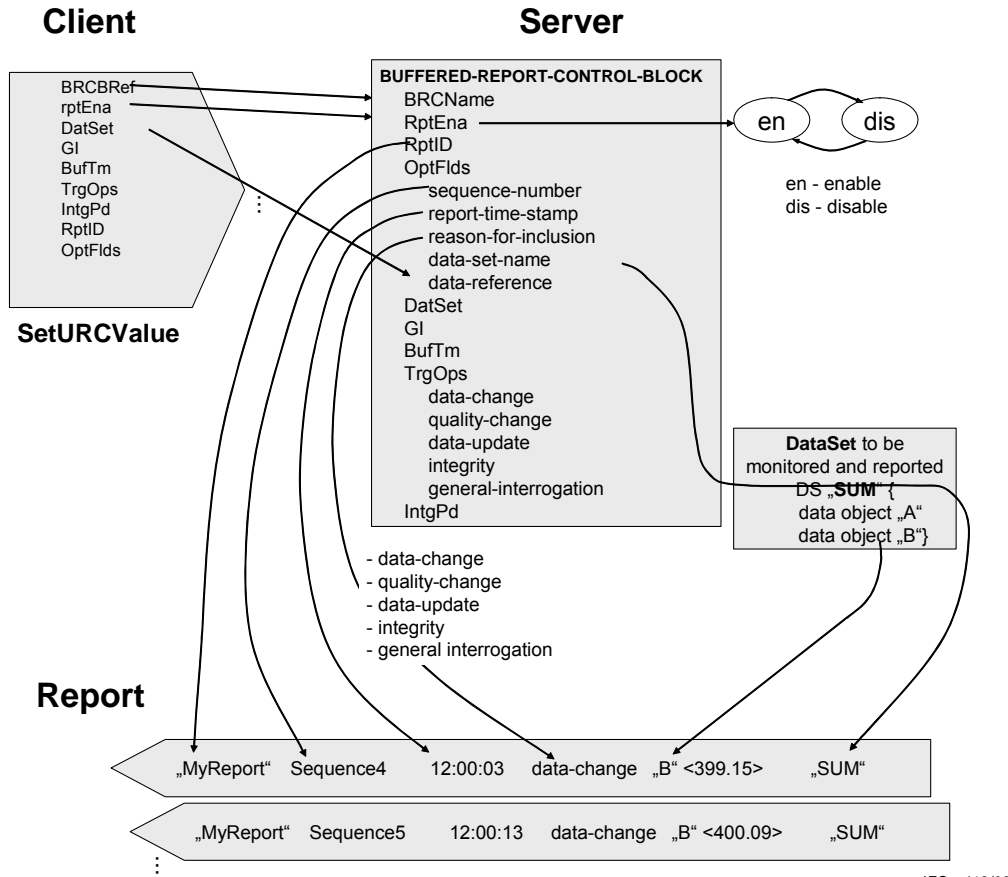
6
7 General-interrogation and integrity reasons for inclusions are mutually exclusive of all rea-
8 sons for inclusion. See 15.2.3.2.3.5.

9 15.2.3.2.3 Procedures for report generation

10 15.2.3.2.3.1 Overview

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

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



IEC 419/03

Figure 30 – Report example

Kommentar [hsf112]: added general-interrogation

1
2

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 **15.2.3.2.3.2 Data-change, quality-change, and data-update**

7 These three trigger options support report generation based on change or update in a value
8 of 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 **DataAttrib-**
13 **ute**.

Kommentar [hsf113]: Changed notation to be more precise.

14 **quality-change**

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

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

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

14 When the **BRCB** is notified by an internal notification of a **data-change**, **quality-change**, or
15 **data-update** event of a member(s) of the referenced **DATA-SET** whose values are to be re-
16 ported, the **BRCB** shall include the value of the member(s) of the referenced **DATA-SET** that
17 produced the internal notification in the report according to 15.2.3.2.2.9. The value to be re-
18 ported shall be the value that produced the internal notification.

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

21 15.2.3.2.3.3 Integrity

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

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

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

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

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

40 15.2.3.2.3.4 GI (General interrogation)

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

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

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

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

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

12 **15.2.3.2.3.5 Time sequence order of reports**

13 The **BRCB** within the implementation resource limits shall send all reports in the time se-
14 quence order in which the related entries have been created.

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

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

28 **15.2.3.2.3.6 Buffering events**

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

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

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

Kommentar [hsf114]: Added

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

41

1 15.2.3.3 GetBRCBValues

2 15.2.3.3.1 GetBRCBValues parameter table

3 A client shall use the **GetBRCBValues** service to retrieve attribute values of **BRCB** made
4 visible and thus accessible to the requesting client by the referenced **LOGICAL-NODE**.

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

7

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

Kommentar [hsf115]: Tissue 49

Kommentar [hsf116]: Tissue 49

8

9 15.2.3.3.2 Request

10 15.2.3.3.2.1 BRCBReference

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

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

13 15.2.3.3.2.2 FunctionalConstraint

14 The parameter **FunctionalConstraint** (**FCCB**) shall contain the value of the functional constraint parameter to filter the respective instances of attributes of a **BRCB**.

16 The service parameter **FunctionalConstraint** (**FCCB**) shall be **BR**.

17 15.2.3.3.3 Response+

18 The parameter **Response+** shall indicate that the service request succeeded. All SCsMs
19 shall be able to return the Response+ parameter values of:

1 **15.2.3.3.3.1 ReportIdentifier**

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

4 **15.2.3.3.3.2 ReportEnable**

5 The parameter **ReportEnable** shall contain the value of the corresponding attribute **RptEna**
6 of the referenced **BRCB**.

7 **15.2.3.3.3.3 DataSetReference**

8 The parameter **DataSetReference** shall contain the value of the corresponding attribute
9 **DatSet** of the referenced **BRCB**.

10 **15.2.3.3.3.4 ConfigurationRevision**

11 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute
12 **ConfRev** of the referenced **BRCB**.

13 **15.2.3.3.3.5 OptionalFields**

14 The parameter **OptionalFields** shall contain the value of the corresponding attribute
15 **OptFlds** of the referenced **BRCB**.

16 **15.2.3.3.3.6 BufferTime**

17 The parameter **BufferTime** shall contain the value of the corresponding attribute **BufTm** of
18 the referenced **BRCB**.

19 **15.2.3.3.3.7 SequenceNumber**

20 The parameter **SequenceNumber** shall contain the value of the corresponding attribute
21 **SqNum** of the referenced **BRCB**.

22 **15.2.3.3.3.8 TriggerOptionsEnabled**

23 The parameter **TriggerOptionsEnabled** shall contain the value of the corresponding attrib-
24 ute **TrgOps** of the referenced **BRCB**.

25 **15.2.3.3.3.9 IntegrityPeriod**

26 The parameter **IntegrityPeriod** shall contain the value of the corresponding attribute
27 **IntgPd** of the referenced **BRCB**.

28 **15.2.3.3.3.10 GeneralInterrogation**

29 The parameter **GeneralInterrogation** shall contain the value of the corresponding attribute
30 **GI** of the referenced **BRCB**

31 **15.2.3.3.3.11 PurgeBuf**

32 The parameter **PurgeBuf** shall contain the value of the corresponding attribute **PurgeBuf** of
33 the referenced **BRCB**.

Kommentar [hsf117]: Tis-
sue 49

34

1 **15.2.3.3.3.12 EntryIdentifier**

2 The parameter **EntryIdentifier** shall contain the value of the corresponding attribute **En-**
 3 **tryID** of the referenced **BRCB**.

4 **15.2.3.3.3.13 TimeOfEntry**

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

Kommentar [hsf118]: Tis-
sue 49

7 **15.2.3.3.3.14 ReserveTimeSecond [0..1]**

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

Kommentar [hsf119]: Tis-
sue 49

10 **15.2.3.3.4 Response–**

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

13 **15.2.3.4 SetBRCBValues**

14 **15.2.3.4.1 SetBRCBValues parameter table**

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

17 NOTE The visible instances are those that are defined within a given view (see Clause 7 for details on
 18 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

19

20 **SetBRCBValues.request** shall be processed atomically by the server. If the request contains
 21 any parameters, whose sets fails, a **Response–** shall be returned.

22

1 **15.2.3.4.2 Request**

2 **15.2.3.4.2.1 BRCBReference**

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

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

5 **15.2.3.4.2.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint** (**FCCB**) shall contain the value of the functional constraint parameter to filter the respective attributes of a **BRCB**.

8 The service parameter **FunctionalConstraint** (**FCCB**) shall be **BR**.

9 **15.2.3.4.2.3 ReportIdentifier [0..1]**

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

12 **15.2.3.4.2.4 ReportEnable [0..1]**

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

15 **15.2.3.4.2.5 DataSetReference [0..1]**

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

18 **15.2.3.4.2.6 OptionalFields [0..1]**

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

21 **15.2.3.4.2.7 BufferTime [0..1]**

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

24 **15.2.3.4.2.8 TriggerOptionsEnabled [0..1]**

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

27 **15.2.3.4.2.9 IntegrityPeriod [0..1]**

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

30 **15.2.3.4.2.10 GeneralInterrogation [0..1]**

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

1 **15.2.3.4.2.11 PurgeBuffer [0..1]**

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

4 **15.2.3.4.2.12 EntryIdentifier[0..1]**

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

7 **15.2.3.4.2.13 ReserveTimeSecond[0..1]**

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

10 **15.2.3.4.3 Response+**

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

12 **15.2.3.4.4 Response–**

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

15

1 **15.2.4 UNBUFFERED-REPORT-CONTROL-BLOCK (URCB) class definition**

2 **15.2.4.1 URCB class Syntax**

3 The **URCB** class shall have the structure defined in Table 25.

4 **Table 28 – 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.					

Kommentar [hsf120]: Tissue 322

Kommentar [hsf121]: Tissue 348

5
6 Except **URCBName**, **URCBRef**, **RptEna**, **SqNum**, and **Resv** all other attributes shall be as
7 defined for the **BRCB** in 14.2.2.

1 **15.2.4.2 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 **15.2.4.3 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 **15.2.4.4 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 **15.2.4.5 Resv – reserve URCB**

18 The attribute **Resv** (if set to TRUE) shall indicate that the **URCB** is currently exclusively re-
19 served for the client that has set the value to TRUE. Other clients shall not be allowed to set
20 any attribute 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 **15.2.5 URCB class services**

25 **15.2.5.1 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 **15.2.5.2 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 OptFids.EntryID, no EntryID shall be included in the Report.
- 2 - regardless of the value of OptFids.BufOvfl, no BufOvfl shall be included in the Report.

Kommentar [hsf122]: Tissue 348

Kommentar [hsf123]: Tissue 348

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.

Kommentar [KS124]: Needs further discussion!

6 **15.2.5.3 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 **15.2.5.4 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

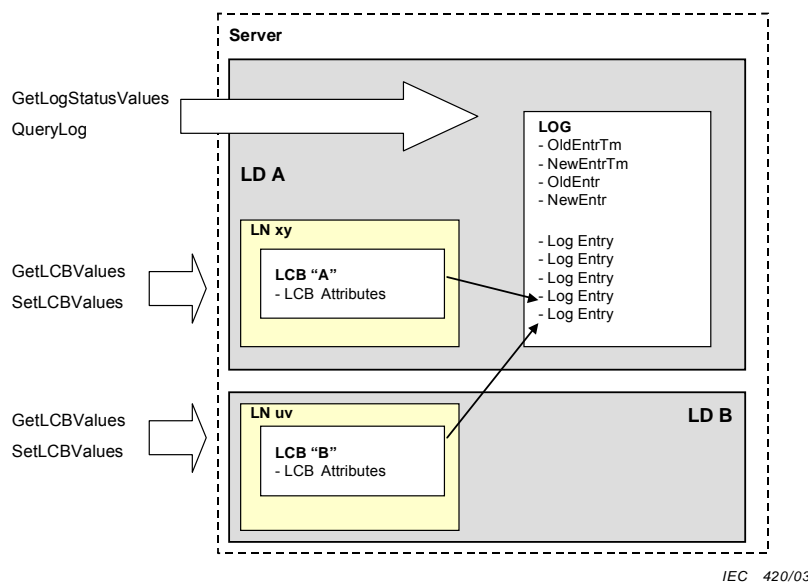
1 15.3 LOG-CONTROL-BLOCK class model

2 15.3.1 General

3 15.3.1.1 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
 8 data 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
 12 over communications.
- 13 – The rate of generation of historical records can in some cases be much faster than the
 14 ability 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
 21 assignment of a sequence number.



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
 25 multiple **LCBs**.

26 15.3.1.2 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
- 4 size of that counter shall be larger than the maximal number of entries that can be stored in a
- 5 **LOG** so that there may not be two entries in the log with the same value of **entryID**. **EntryID**
- 6 together with **TimeOfEntry** provide a unique identification of the entry.

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

8 **15.3.2 LCB class definition**

9 **15.3.2.1 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**.
 12 Changes in a value of a member of a **DATA-SET** shall be stored as **LOG** entry. Multiple **LCBs**
 13 allow multiple **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 29.

18 **Table 29 – LCB class definition**

LCB class				
Attribute name	Attribute type	FC	TrgOp	Value/value range/explanation
LCBName	ObjectName	-	-	Instance name of an 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			
BufTm	INT32U	LG	dchg	
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				

Kommentar [KS125]: Due to editors' meeting in Zug.

- 19
- 20 **15.3.2.2 LCB class attributes**

21 **15.3.2.2.1 LCBName – log control name**

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

1 **15.3.2.2.2 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 **15.3.2.2.3 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
9 log 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 **15.3.2.2.4 DataSet – data set reference**

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

15 **15.3.2.3 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 **En-**
18 **tryData** (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 **15.3.2.3.1 BufTm – buffer time**

23 This attribute shall be the same as defined in 15.2.2.9. |

Kommentar [KS126]: In-
cluded as decided in Zug

24 **15.3.2.3.2 TrgOps – trigger options**

25 The attribute **TrgOps** shall specify the trigger conditions that shall be monitored by this **LCB**
26 to cause a Log entry to be created. The values defined are the same as for reporting (see
27 14.2.2.11).

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

29 **15.3.2.3.3 IntgPd – integrity period**

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

1 **15.3.2.3.4 LogRef – log reference**

2 The attribute **LogRef** shall be the reference of the **LOG** to which values of members of the
3 referenced **DATA-SET** shall be recorded.

4 **15.3.2.4 LCB services – Overview**

5 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

6

7 **15.3.2.5 GetLCBValues**

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

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

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

12

13 **15.3.2.5.1 Request**

14 **15.3.2.5.1.1 LCBReference**

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

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

17 **15.3.2.5.1.2 FunctionalConstraint**

18 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional con-
19 straint parameter to filter the respective instances of attributes of a **LCB**.

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

21 **15.3.2.5.2 Response+**

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

1 **15.3.2.5.2.1 LogEnable**

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

4 **15.3.2.5.2.2 DataSetReference**

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

7 **15.3.2.5.2.3 TriggerOptions**

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

10 **15.3.2.5.2.4 IntegrityPeriod**

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

13 **15.3.2.5.2.5 LogReference**

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

16 **15.3.2.5.3 Response–**

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

19 **15.3.2.6 SetLCBValues**

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

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

24

25 **SetLCBValues.request** shall be processed atomically by the server. If the request contains
26 any parameters, whose sets fails, a **Response–** shall be returned.

- 1
- 2 **15.3.2.6.1 Request**
- 3 **15.3.2.6.1.1 LCBReference**
- 4 The parameter **LCBReference** shall specify the **ObjectReference** of **LCB**.
- 5 The service parameter **LCBReference** shall be **LCBRef**.
- 6 **15.3.2.6.1.2 FunctionalConstraint**
- 7 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional con-
- 8 straint parameter to filter the respective instances of attributes of a **LCB**.
- 9 The service parameter **FunctionalConstraint (FCCB)** shall be **LG** (logging).
- 10 **15.3.2.6.1.3 LogEnable [0..1]**
- 11 The parameter **LogEnable** shall contain the value of the corresponding attribute **LogEna** of
- 12 the referenced **LCB**.
- 13 **15.3.2.6.1.4 DataSetReference [0..1]**
- 14 The parameter **DataSetReference** shall contain the value of the corresponding attribute
- 15 **DatSet** of the referenced **LCB**.
- 16 **15.3.2.6.1.5 OptionalFields [0..1]**
- 17 The parameter **OptionalFields** shall contain the value of the corresponding attribute
- 18 **OptFlds** of the referenced **LCB**.
- 19 **15.3.2.6.1.6 IntegrityPeriod [0..1]**
- 20 The parameter **IntegrityPeriod** shall contain the value of the corresponding attribute
- 21 **IntgPd** of the referenced **LCB**.
- 22 **15.3.2.6.1.7 LogReference [0..1]**
- 23 The parameter **LogReference** shall contain the value of the corresponding attribute **LogRef**
- 24 of the referenced **LCB**.
- 25 **15.3.2.6.2 Response+**
- 26 The parameter **Response+** shall indicate that the service request succeeded.
- 27 **15.3.2.6.3 Response-**
- 28 The parameter **Response-** shall indicate that the service request failed. The appropriate
- 29 **ServiceError** shall be returned.
- 30 This service shall return a failure if the service has been issued for any attribute of a **LCB**
- 31 other than enable while **LCB** is enabled.

Kommentar [hsf127]: fixed

1 **15.3.3 LOG class definition**

2 **15.3.3.1 LOG class syntax**

3 The LOG shall be filled on a first-in first-out basis. When the list of log entries reaches a point
 4 where the stored data reaches the maximal size of the log, the oldest log entry shall be over-
 5 written. This action shall have no impact to the further incrementing of the **EntryID** of the
 6 added log entries.

7 The LOG shall have the structure defined in Table 30.

8 **Table 30 – 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 15.2.3.2.2.10.
Services			
QueryLogByTime QueryLogAfter GetLogStatusValues			

Kommentar [hsf128]: tis-sue 339

9

10 **15.3.3.2 LOG class attributes**

11 **15.3.3.2.1 LogName – log name**

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

14 **15.3.3.2.2 LogRef – log reference**

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

16 The **ObjectReference LogRef** shall be:

LDName/LNName.LogName

Kommentar [KS129]: As discussed in San Diego 2007-02

17

1 **15.3.3.2.3 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 en-
4 try itself, which indicates when the event that caused the creation of the log entry has occurred.

5 **15.3.3.2.4 NewEntrTm – newest log entry time of log**

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

7 **15.3.3.2.5 OldEntr – oldest log entry sequence number**

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

9 **15.3.3.2.6 NewEntr – newest log entry sequence number**

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

11 **15.3.3.2.7 Entry [1..n]**

12 **15.3.3.2.7.1 TimeOfEntry – time of log entry**

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

16 **15.3.3.2.7.2 EntryID – entry identifier**

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

19 **15.3.3.2.7.3 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 val-
22 ues of all data attributes of the member of **DATA-SET**.

23 **DataRef**

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

Kommentar [KS130]: Re-
moved one "contain"

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**:

Kommentar [hsf131]: cha
nged from report to log

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**
35 **integrity** (=TRUE) all values of all members of a **DATA-SET** shall be included in the
36 log 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**
 4 that caused the creation of the report. The value range for reasons for inclusion shall
 5 be 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 15.3.4 Procedures to generate the log entries

11 15.3.4.1 Overview

12 Basically, the conditions and constraints for log generation shall be the same as for report
 13 generation (see 14.2.3.2.3). Subclause 14.3.4 specifies the differences only.

14 15.3.4.2 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 15.3.4.3 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 15.3.5 LOG services

22 15.3.5.1 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 15.3.5.2 QueryLogByTime

26 15.3.5.2.1 QueryLogByTime parameter table

27 A client shall use the **QueryLogByTime** service to retrieve a range of **LOG** entries from a
 28 **LOG** based on time ranges (**RangeStartTime** and **RangeStopTime**).

Parameter name
Request
LogReference
RangeStartTime
RangeStopTime
Response+
ListOfLogEntries
Response-
ServiceError

29

1 **15.3.5.2.2 Request**

2 **15.3.5.2.2.1 LogReference**

3 The parameter **LogReference** shall contain the **ObjectReference LogRef** of a **LOG**. The
4 **ObjectReference LogReference** shall be:

LDName/LNName.LogName

Kommentar [KS132]: As discussed in San Diego 2007-02

5 **15.3.5.2.2.2 RangeStartTime**

6 The parameter **RangeStartTime** shall contain the range start time to retrieve log entries.
7 The first log entry selected shall be the first entry in the log with a **RangeStartTime** greater
8 than, 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 **15.3.5.2.2.3 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 **15.3.5.2.3 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 re-
19 quest.

20 **15.3.5.2.4 Response-**

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

23 **15.3.5.3 QueryLogAfter**

24 **15.3.5.3.1 QueryLogAfter parameter table**

25 A client shall use the **QueryLogAfter** service to retrieve a range of **LOG** entries from the ref-
26 erenced **LOG** based on ranges of IDs that are after the **RangeStartTime** and **Entry**.

Parameter Name
Request
LogReference
RangeStartTime
Entry
Response+
ListOfLogEntries
Response-
ServiceError

27

1 **15.3.5.3.2 Request**2 **15.3.5.3.2.1 LogReference**

3 The parameter **LogReference** shall specify the **ObjectReference LogRef** of the **LOG**. The
 4 **ObjectReference LogReference** shall be:

LDName/LNName.LogName

Kommentar [KS133]: As discussed in San Diego 2007-02

5

6 **15.3.5.3.2.2 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 **15.3.5.3.2.3 Entry**

10 The parameter **Entry** shall reference the **LOG** entry of the selected **RangeStartTime** after
 11 which the log entries shall be selected.

12 **15.3.5.3.3 Response+**13 **15.3.5.3.3.1 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 **15.3.5.3.4 Response–**

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

19 **15.3.5.4 GetLogStatusValues**

20 A client shall use the **GetLogStatusValues** service to retrieve the attribute values of a **LOG**
 21 made visible and thus accessible to the requesting client by the referenced **LLNO**.

Kommentar [KS134]: #52
 1 changed name GetLCBValues to GetLogStatusValues

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 **15.3.5.4.1 Request**

2 **15.3.5.4.1.1 LogReference**

3 The parameter **LogReference** shall specify the **ObjectReference** of the **LOG**.

4 The service parameter **LogReference** shall be:

LDName/LNName.LogName

Kommentar [KS135]: As discussed in San Diego 2007-02

5 **15.3.5.4.1.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint** (**FCCB**) shall contain the value of the functional constraint parameter to filter the respective instances of attributes of a **LOG**.

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

9 **15.3.5.4.2 Response+**

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

11 **15.3.5.4.2.1 OldestEntryTime**

12 The parameter **OldestEntryTime** shall contain the value for the corresponding attribute **OldEntrTm** of the referenced **LOG**.

14 **15.3.5.4.2.2 NewestEntryTime**

15 The parameter **NewestEntryTime** shall contain the value for the corresponding attribute **NewEntrTm** of the referenced **LOG**.

17 **15.3.5.4.2.3 OldestEntry**

18 The parameter **OldestEntry** shall contain the value for the corresponding attribute **OldEntr** of the referenced **LOG**.

20 **15.3.5.4.2.4 NewestEntry**

21 The parameter **NewestEntry** shall contain the value for the corresponding attribute **NewEntr** of the referenced **LOG**.

23 **15.3.5.4.3 Response-**

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

26

1 **16 Generic substation event class model (GSE)**

2 **16.1 Overview**

3 The generic substation event model provides the possibility for a fast and reliable system-
 4 wide distribution of input and output data values. The generic substation event model is
 5 based on the concept of an autonomous decentralization, providing an efficient method allow-
 6 ing the simultaneous delivery of the same generic substation event information to more than
 7 one physical 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
 21 writes the values in a local buffer at the sending side; the receiver reads the values from a
 22 local buffer at the receiving side. The communication system is responsible to update the lo-
 23 cal buffers of the subscribers. A generic substation event control class in the publisher is
 24 used to control the procedure.

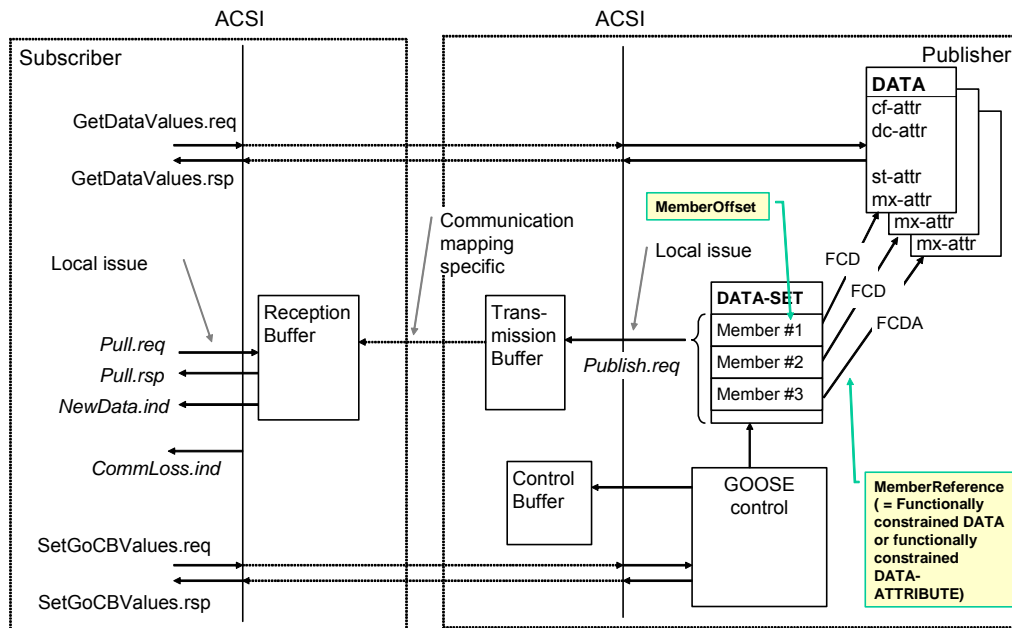


Figure 32 – GoCB model

IEC 421/03

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**
 4 changes, the transmission buffer of the publisher is updated with the local service "publish"
 5 and the values are transmitted with a **GOOSE** message. The **DATA-SET** may have several
 6 members (numbered from 1 up – the numbers shall be called **MemberOffset**). Each member
 7 shall have a **MemberReference** referencing the **DataAttribute** with a specific functional
 8 constraint (**FC**). Mapping specific services of the communication network will update the con-
 9 tent of the buffer in the subscribers. New values received in the reception buffer are signalled
 10 to the application.

Kommentar [CB136]: TIS SUE #33

11 The **GOOSE** messages contain information that allow the receiving device to know that a
 12 status has changed and the time of the last status change. The time of the last status change
 13 allows a receiving device to set local timers relating to a given event.

14 A newly activated device, upon power-up or reinstatement to service, shall send current data
 15 (status) or values as the initial **GOOSE** message. Moreover, all devices sending **GOOSE** mes-
 16 sages shall continue to send the message with a long cycle time, even if no status/value
 17 change has occurred. This ensures that devices that have been activated recently will know
 18 the current status values of their peer devices.

19 NOTE 3 The **GSSE** model is similar to the **GOOSE** model. The basic concept described above applies also to the
 20 **GSSE** model. One major difference is the kind of information exchanged. **GOOSE** provides a flexible means to
 21 specify which information is to be exchanged (**DATA-SET**) whereas **GSSE** provides a simple list of status informa-
 22 tion.

23 The behaviour of the **GoCB** shall apply to the **GsCB**.

24 **16.2 GOOSE-CONTROL-BLOCK (GoCB) class**

25 **16.2.1 GoCB definition**

26 The **GoCB** shall be as defined in Table 31.

27 **Table 31 – 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	GO	dchg	
ConfRev	INT32U	GO	dchg	
NdsCom	BOOLEAN	GO	dchg	
DstAddress	PHYCOMADDR	GO	dchg	
Services				
SendGOOSEMessage				
GetGoReference				
GetGOOSEElementNumber				
GetGoCBValues				
SetGoCBValues				

Kommentar [CB137]: TIS SUE #38

Kommentar [CB138]: Harmonization with sampled values

Kommentar [CB139]: TIS SUE #39 – Data type according to TISSUE #142

28

29 **16.2.1.1 GoCBName – GOOSE control block name**

30 The attribute **GoCBName** shall unambiguously identify a **GoCB** within the scope of a **LLNO**.

1 16.2.1.2 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 16.2.1.3 GoEna – GOOSE enable

5 The attribute **GoEna** (if set to TRUE) shall indicate that the **GoCB** is currently enabled to
6 send 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
8 the value of **ConfRev** equals 0, a **SetGoCBValues** with the parameter **GoEna** equals TRUE
9 shall fail and a negative response shall be issued.

Kommentar [CB140]: TIS
SUE #333

10 While being TRUE (**GoCB** enabled), no changes of attribute values of the **GoCB** other than
11 disabling shall be allowed.

12 16.2.1.4 GoID – GOOSE identifier

13 The attribute **GoID** shall be a user definable identification of the GOOSE message. The de-
14 fault value shall be that of the **ObjectReference GoCB**. However, the value may be set to an-
15 other 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 16.2.1.5 DataSet – 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 16.2.1.6 ConfRev – configuration revision

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

- 33 – any deletion of a member of the **DATA-SET**;
- 34 – the reordering of members of the **DATA-SET**; and
- 35 – changing the value of the attribute **DatSet**.

36 The counter shall be incremented when the configuration changes. At configuration time, the
37 configuration tool will be responsible for incrementing/maintaining the **ConfRev** value. When
38 configuration changes occur due to **SetGoCBValues**, the IED shall be responsible for incre-
39 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 **16.2.1.7 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 **16.2.1.8 DstAddress**

12 The attribute **DstAddress** shall be the SCSM specific addressing information like media ac-
13 ccess address, priority, and other information.

Kommentar [CB141]: TIS SUE #39

14 **16.2.2 GOOSE service Definitions**

15 **16.2.2.1 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

Kommentar [CB142]: TIS SUE #36

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 **16.2.2.2 SendGOOSEMessage**

22 **16.2.2.2.1 SendGOOSEMessage parameter table**

23 The **SendGOOSEMessage** service shall be used by a **GoCB** to send a **GOOSE** message
24 over a **MULTICAST-APPLICATION-ASSOCIATION**.

Parameter name
Request
GOOSE message

25

1 **16.2.2.2.2 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 **16.2.2.3 GetGoReference**6 **16.2.2.3.1 GetGoReference parameter table**

7 A client shall use the **GetGoReference** service to retrieve the **MemberReferences** of spe-
8 cific 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 **16.2.2.3.2 Request**11 **16.2.2.3.2.1 GoCBReference**

12 The parameter **GoCBReference** shall identify the attribute **GoCBRef** of the **GoCB** for which
13 **MemberReferences** are being requested.

14 **16.2.2.3.2.2 MemberOffset [1..n]**

15 The parameter **MemberOffset** shall contain a number identifying a member of the **DATA-**
16 **SET** referenced by the attribute **DatSet**.

17 **16.2.2.3.3 Response+**18 **16.2.2.3.3.1 GoCBReference**

19 The parameter **GoCBReference** shall contain the parameter that identifies the attribute
20 **GoCBRef** of the **GoCB** for which **MemberReferences** are returned.

21 **16.2.2.3.3.2 ConfigurationRevision**

22 The parameter **ConfigurationRevision** shall contain the attribute **ConfRev** of the **GoCB**.

23 **16.2.2.3.3.3 DatSet**

24 The parameter **DatSet** shall contain the attribute **DatSet** of the **GoCB**.

Kommentar [CB143]: TIS
SUE #36

1 **16.2.2.3.3.4 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 **16.2.2.3.4 Response-**

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

9 **16.2.2.4 GetGOOSEElementNumber**

10 **16.2.2.4.1 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**.

Kommentar [CB144]: TIS SUE #36 claims that the **DatSetReference** is missing as a parameter – however, I disagree with that, since the **GoCBReference** is sufficient.

Parameter name
Request
GoCBReference
MemberReference [1..n]
Response+
GoCBReference
ConfigurationRevision
DatSet
MemberOffset [1..n]
Response-
ServiceError

13

14 **16.2.2.4.2 Request**

15 **16.2.2.4.2.1 GoCBReference**

16 The parameter **GoCBReference** shall identify the attribute **GoCBRef** of the **GoCB** for which
17 **MemberOffset** are being requested.

18 **16.2.2.4.2.2 MemberReference [1..n]**

19 The parameter **MemberReference** shall contain the **MemberReference** for which the
20 **MemberOffset** of a member of the **DATA-SET** is requested. A value of NULL is reserved to
21 indicate that no member of the referenced **DATA-SET** is defined for the member being re-
22 quested with a **MemberReference**.

23 **16.2.2.4.3 Response+**

24 **16.2.2.4.3.1 GoCBReference**

25 The parameter **GoCBReference** shall contain the parameter that identifies the attribute
26 **GoCBRef** of the **GoCB** for which **MemberOffsets** are returned.

27 **16.2.2.4.3.2 ConfigurationRevision**

28 The parameter **ConfigurationRevision** shall contain the attribute **ConfRev** of the **GoCB**.

1 **16.2.2.4.3.3** **DatSet**2 The parameter **DatSet** shall contain the attribute **DatSet** of the **GoCB**.3 **16.2.2.4.3.4** **MemberOffset [1..n]**4 The parameter **MemberOffset** shall contain the **MemberOffset** requested for the **MemberReference** of a member of the **DATA-SET**. A value of **NULL** shall indicate that no member of the referenced **DATA-SET** is defined matching with a **MemberReference**.7 **16.2.2.4.4** **Response–**8 The parameter **Response–** shall indicate that the service request failed. The appropriate **ServiceError** shall be returned.10 **16.2.2.5** **GetGoCBValues**11 A client shall use the **GetGoCBValues** service to retrieve attribute values of **GoCB** made visible and thus accessible to the requesting client by the referenced **LLNO**.

13 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
GoCBReference
FunctionalConstraint
Response+
GoEnable
GOOSEID
DataSetReference
ConfigurationRevision
NeedsCommissioning
Response–
ServiceError

15

16 **16.2.2.5.1** **Request**17 **16.2.2.5.1.1** **GoCBReference**18 The parameter **GoCBReference** shall specify the **ObjectReference** of the **GoCB**.19 The service parameter **LCBReference** shall be **LDName/LLNO.GoCBName**.20 **16.2.2.5.1.2** **FunctionalConstraint**21 The parameter **FunctionalConstraint** (**FCCB**) shall contain the value of the functional constraint parameter to filter the respective instances of attributes of a **GoCB**.23 The service parameter **FunctionalConstraint** (**FCCB**) shall be **GO** (goose control).24 **16.2.2.5.2** **Response+**25 The parameter **Response+** shall indicate that the service request succeeded.

1 **16.2.2.5.2.1 GoEnable**

2 The parameter **GoEnable** shall contain the value of the corresponding attribute **GoEna** of the
3 referenced **GoCB**.

4 **16.2.2.5.2.2 GOOSEID**

5 The parameter **GOOSEID** shall contain the value of the corresponding attribute **GoID** of the
6 referenced **GoCB**.

7 **16.2.2.5.2.3 DataSetReference**

8 The parameter **DataSetReference** shall contain the value of the corresponding attribute
9 **DatSet** of the referenced **GoCB**.

10 **16.2.2.5.2.4 ConfigurationRevision**

11 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute
12 **ConfRev** of the **GoCB**.

13 **16.2.2.5.2.5 NeedsCommissioning**

14 The parameter **NeedsCommissioning** shall contain the value of the corresponding attribute
15 **NdsCom** of the **GoCB**.

16 **16.2.2.5.3 Response-**

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

19 **16.2.2.6 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 **16.2.2.6.1 Request**

27 **16.2.2.6.1.1 GoCBReference**

28 The parameter **GoCBReference** shall specify the **ObjectReference** of the **GoCB**.

1 The service parameter **GoCBReference** shall be **LDName/LLNO.GoCBName**.

2 **16.2.2.6.1.2 FunctionalConstraint**

3 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional con-
4 straint parameter to filter the respective instances of attributes of a **GoCB**.

5 The service parameter **FunctionalConstraint (FCCB)** shall be **GO** (goose control).

6 **16.2.2.6.1.3 GoEnable [0..1]**

7 The parameter **GoEnable** shall contain the value for the corresponding attribute **GoEna** of
8 the referenced **GoCB**.

9 **16.2.2.6.1.4 GOOSEID [0..1]**

10 The parameter **GOOSEID** shall contain the value for the corresponding attribute **GoID** of the
11 referenced **GoCB**.

12 **16.2.2.6.1.5 DataSetReference [0..1]**

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

15 **16.2.2.6.2 Response+**

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

17 **16.2.2.6.3 Response–**

18 The parameter **Response–** shall indicate that the service request failed. The appropriate
19 **ServiceError** 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 **16.2.3 Generic object oriented substation event (GOOSE) message**

23 **16.2.3.1 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 32.

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 32 – 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

Kommentar [CB146]: TIS SUE #40

Kommentar [CB147]: TIS SUE #37

2

3 **16.2.3.2 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 **16.2.3.3 GoID – application identifier**

7 The parameter **GoID** shall contain the value of the attribute **GoID** of the **GoCB**.

8 **16.2.3.4 GoCBRef – GOOSE control block reference**

9 The parameter **GoCBRef** shall contain the reference of the **GOOSE** control block.

10 **16.2.3.5 T – time stamp**

11 The parameter **T** shall contain the time at which the attribute **StNum** was incremented.

12 **16.2.3.6 StNum – state number**

13 The parameter **StNum** shall contain the counter that increments each time a **GOOSE** mes-

14 sage has been sent and a value change has been detected within the **DATA-SET** specified

15 by **DatSet**.

16 The initial value for **StNum** upon a transition of **GoEna** to **TRUE** shall be 1. The value of 0

17 shall be reserved.

18 NOTE – a transition of **GoEna** to **TRUE** happens directly at power-up or restart for properly and consistently con-

19 figured **GoCB**.

20 **16.2.3.7 SqNum – sequence number**

21 The parameter **SqNum** shall contain the counter that shall increment each time a **GOOSE**

22 message has been sent.

23 Following a **StNum** change, the counter **SqNum** shall be set to a value of 0. If the counter

24 **SqNum** overruns, it shall be set to a value of 1. The initial value for **SqNum** upon a transition

25 of **GoEna** to **TRUE** shall be 1.

Kommentar [CB148]: TIS SUE #52

1 **16.2.3.8 Test – test**

2 The parameter **Test** shall indicate with the value of TRUE that the values of the message
3 shall not be used for operational purposes.

4 **16.2.3.9 ConfRev – configuration revision**

5 The parameter **ConfRev** (taken from the **GoCB**) shall contain the count of the number of
6 times that the configuration of the **DATA-SET** referenced by **DatSet** has been changed.

7 **16.2.3.10 NdsCom – needs commissioning**

8 The parameter **NdsCom** shall contain the attribute **NdsCom** (taken from the **GoCB**) of the
9 **GoCB**.

10 **16.2.3.11 GOOSEData [1..n]**

11 The parameter **GOOSEData** shall contain the user-defined information (of the members of
12 **DATA-SET**) to be included in a **GOOSE** message.

13 The parameter **Value** shall contain the value of a member of the **DATA-SET** referenced in
14 the **GoCB**.

15

1 **17 Transmission of sampled value class model**

2 **17.1 Overview**

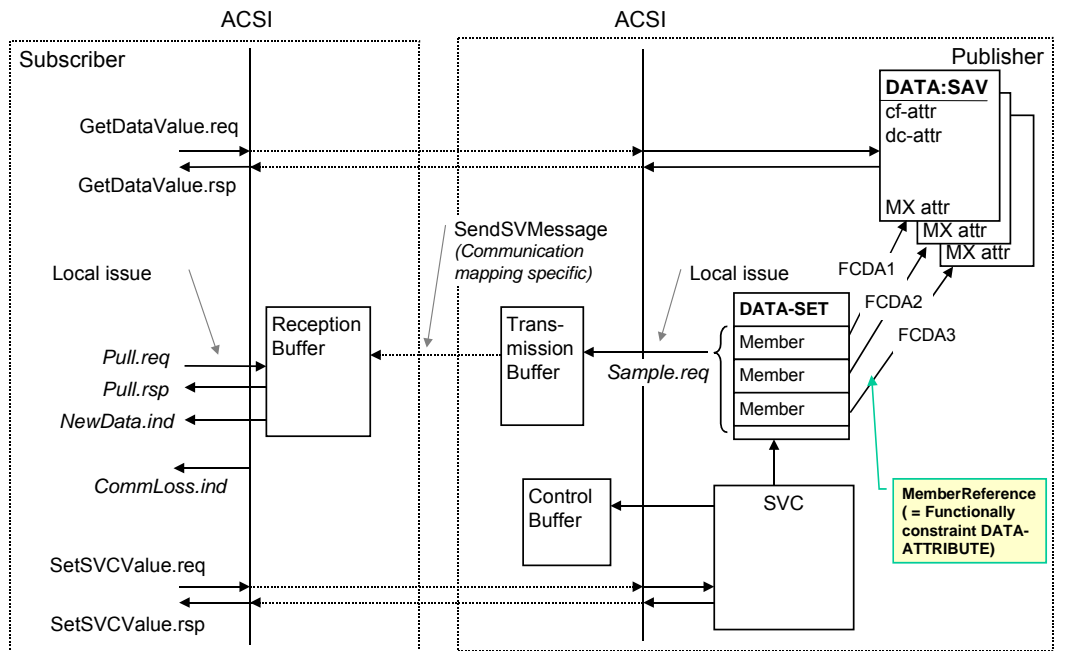
3 The transmission of sampled values requires special attention with regard to the time con-
4 straints. The model provides transmission of sampled values in an organized and time-
5 controlled way so that the combined jitter of sampling and transmission is minimized to a degree
6 that an unambiguous 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**
8 are of the common data class SAV (sampled value as defined in IEC 61850-7-3). A buffer
9 structure shall be defined for the transmission of the sampled values.

10 The information exchange shall be based on a publisher/subscriber mechanism. The pub-
11 lisher shall write the values in a local buffer at the sending side; the subscriber shall read the
12 values from a local buffer at the receiving side. A time stamp shall be added to the values, so
13 that the subscriber can check the timeliness of the values. The communication system shall
14 be responsible to update the local buffers of the subscribers. A sampled value control (**SVC**)
15 in the publisher shall be used to control the communication procedure.

16 Figure 33 gives an overview on the classes and services of the model.

17



IEC 423/03

Figure 33 – Model for transmission of sampled values

21 There shall be two methods to exchange sampled values between a publisher and one or
22 more subscriber. One method shall use the **MULTICAST-APPLICATION-ASSOCIATION**
23 (**multicast 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
 3 placed into the receive buffers of the subscribers. The arrival of a new series of samples in
 4 the receive 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 **17.2 Transmission of sampled values using multicast**

9 The transmission of sampled values using multicast (**MULTICAST-SAMPLE-VALUE-CONTROL-BLOCK – MSVCB**) shall be based on configured configuration in the producer de-
 10 vice. The data exchange shall be based on the multicast application association. To support
 11 self-descriptive capabilities, any client may read the attributes of the sampled value control
 12 instance. Authorized clients may modify attributes of the sampled value control.
 13

14 **17.2.1 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	MS	dchg	
ConfRev	INT32U	MS	dchg	
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				

Kommentar [KS149]: #18
8

17 **17.2.1.1 MsvCBNam – multicast sampled value control name**

18 The attribute **MsvCBNam** shall unambiguously identify a **MSVCB** within the scope of an
 19 **LLNO**.

20 **17.2.1.2 MsvCBRef – multicast sampled value control reference**

21 The attribute **MsvCBRef** shall be the unique path-name of a **MSVCB** within an **LLNO**.

22 The **ObjectReference MsvCBRef** shall be:

LDName/LLN0.MsvCBNam

1

2 **17.2.1.3 SvEna – sampled value enable**

3 The attribute **SvEna** (if set to TRUE) shall indicate that the **MSVCB** is currently enabled to
4 send 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
6 than disabling shall be allowed.

7 **17.2.1.4 MsvID – multicast sampled value identifier**

8 The attribute **MSVID** shall be a unique identification of the sampled value buffer related to
9 the update of the sampled values.

10 **17.2.1.5 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 **17.2.1.6 ConfRev – configuration revision**

14 The attribute **ConfRev** shall contain a count of the number of times that the configuration
15 with 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 **DATA-SET** whose functional con-
19 straint 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
24 reserved. 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 **ConfRev** are those made by local means like system configura-
27 tion.

28 **17.2.1.7 SmpRate**

29 The attribute **SmpRate** shall specify the sample rate in units of samples per nominal period.

30 **17.2.1.8 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**
32 message issued by this **MSVCB**. This attribute defines a subset of the optional header fields
33 that 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)

Kommentar [KS150]: #18
8

1 17.2.2 Multicast sampled value class services

2 17.2.2.1 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 17.2.2.2 SendMSVMessage

6 17.2.2.2.1 SendMSVMessage parameter table

7 The **SendMSVMessage** service shall be used by an **MSVCB** to send sampled values from
8 the server to the client over a **MULTICAST-APPLICATION-ASSOCIATION**.

Parameter name
Request
MSV message

9

10 17.2.2.2.2 Request

11 17.2.2.2.2.1 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 17.2.2.3 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

Kommentar [KS151]: #18
7

20

1 **17.2.2.3.1 Request**

2 **17.2.2.3.1.1 MsvCBReference**

3 The parameter **MsvCBReference** shall specify the **ObjectReference** of the **MSVCB**.

4 The service parameter **MsvCBReference** shall be **LDName/LLNO.MsvCBNam**.

5 **17.2.2.3.1.2 FunctionalConstraint**

6 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional constraint parameter to filter the respective instances of attributes of a **MSVCB**.

8 The service parameter **FunctionalConstraint (FCCB)** shall be **MS** (multicast sampled value control).

10 **17.2.2.3.2 Response+**

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

12 **17.2.2.3.2.1 SvEnable**

13 The parameter **SvEnable** shall contain the value of the corresponding attribute **SvEna** of the referenced **MSVCB**.

15 **17.2.2.3.2.2 MulticastSampleValueID**

16 The parameter **MulticastSampleValueID** shall contain the value of the corresponding attribute **MsvID** of the referenced **MSVCB**.

18 **17.2.2.3.2.3 DataSetReference**

19 The parameter **DataSetReference** shall contain the value of the corresponding attribute **DatSet** of the referenced **MSVCB**.

21 **17.2.2.3.2.4 ConfigurationRevision**

22 The parameter **ConfigurationRevision** shall contain the value of the corresponding attribute **ConfRev** of the **MSVCB**.

24 **17.2.2.3.2.5 SampleRate**

25 The parameter **SampleRate** shall contain the value of the corresponding attribute **SmpRate** of the **MSVCB**.

27 **17.2.2.3.2.6 OptionalFields**

28 The parameter **OptFlds** shall contain the value of the corresponding attribute **OptFlds** of the **MSVCB**.

Kommentar [KS152]: #18
7

30 **17.2.2.3.3 Response-**

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

33 **17.2.2.4 SetMSVCBValues**

34 A client shall use the **SetMSVCB Values** service to set attribute values of **MSVCB** 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).

3

Parameter name
Request
MsvCBReference
FunctionalConstraint
SvEnable [0..1]
MulticastSampleValueID [0..1]
DataSetReference [0..1]
SampleRate [0..1]
OptFlds [0..1]
Response+
Response-
ServiceError

Kommentar [KS153]: #18
7

4

5 17.2.2.4.1 Request

6 17.2.2.4.1.1 MsvCBReference

7 The parameter **MsvCBReference** shall specify the **ObjectReference** of the **MSVCB**.

8 The service parameter **MsvCBReference** shall be **LDName/LLNO.MsvCBNam**.

9 17.2.2.4.1.2 FunctionalConstraint

10 The parameter **FunctionalConstraint** (**FCCB**) shall contain the value of the functional con-
11 straint parameter to filter the respective instances of attributes of a **MSVCB**.

12 The service parameter **FunctionalConstraint** (**FCCB**) shall be **MS** (multicast sampled value
13 control).

14 17.2.2.4.1.3 SvEnable [0..1]

15 The parameter **SvEnable** shall contain the value for the corresponding attribute **SvEna** of the
16 referenced **MSVCB**.

17 17.2.2.4.1.4 MulticastSampleValueID [0..1]

18 The parameter **MulticastSampleValueID** shall contain the value for the corresponding at-
19 tribute **MsvID** of the referenced **MSVCB**.

20 17.2.2.4.1.5 DataSetReference [0..1]

21 The parameter **DataSetReference** shall contain the value for the corresponding attribute
22 **DatSet** of the referenced **MSVCB**.

23 17.2.2.4.1.6 SampleRate [0..1]

24 The parameter **SampleRate** shall contain the value for the corresponding attribute **SmpRate**
25 of the **MSVCB**.

1 **17.2.2.4.1.7 OptionalFields**

2 The parameter **OptFlds** shall contain the value of the corresponding attribute **OptFlds** of the
3 **MSVCB**.

Kommentar [KS154]: #18
7

4 **17.2.2.4.2 Response+**

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

6 **17.2.2.4.3 Response-**

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

9 This service shall return a failure if the service has been issued for any attribute of a **MSVCB**
10 other than **SvEnable** while **MSVCB** is enabled.

11 **17.3 Transmission of sampled values using unicast**

12 The transmission of sampled values using unicast (**UNICAST-SAMPLE-VALUE-CONTROL-**
13 **BLOCK – USVCB**) shall be based on two-party application associations. The subscriber shall
14 establish the association with the producer. The subscriber may then configure the class and
15 enable the transmission of the sampled values with the attribute **SvEna**. When the associa-
16 tion is released, the transmission of the sampled values shall stop and the instance of the
17 control class shall be released.

18 The samples shall be sent using the two-party application association.

19 **17.3.1 USVCB class definition**

20 The **USVCB** shall be as defined in Table 34.

21 **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	INT32U	US	dchg	
SmpRate	INT16U	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				

Kommentar [KS155]: #18
8

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

17.3.1.1 UsvCBNam – unicast sampled value control name

The attribute **UsvCBNam** shall unambiguously identify a **USVCB** within the scope of a **LLNO**.

17.3.1.2 UsvCBRef – unicast sampled value control reference

The attribute **UsvCBRef** shall be the unique path-name of a **USVCB** within a **LLNO**.

The **ObjectReference UsvCBRef** shall be:

LDName/LLNO.UsvCBNam

17.3.1.3 SvEna – sampled value enable

The attribute **SvEna** (if set to TRUE) shall indicate that the **USVCB** is currently enabled to send values of the **USVCB**. If set to FALSE the **USVCB** shall stop issuing reports.

While being TRUE (**USVCB** enabled), no changes of attribute values of **USVCB** other than disabling shall be allowed.

If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client that has enabled the **USVCB** is lost, the **USVCB** shall set the attribute to FALSE.

17.3.1.4 Resv – reserve USVCB

The attribute **Resv** (if set to TRUE) shall indicate that the **USVCB** is currently exclusively reserved for the client that has set the value to TRUE. Other clients shall not be allowed to set any attribute of that **USVCB**.

If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client that has set this attribute to TRUE is lost, the **USVCB** shall set the attribute to FALSE.

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

17.3.1.5 UsvID

The attribute **UsvID** shall be a unique identification of the sampled value buffer related to the update of the sampled values.

17.3.1.6 DatSet

The attribute **DatRef** shall specify the reference of the **DATA-SET** whose values of members are to be transmitted in the **USVCB** message.

17.3.1.7 ConfRev – configuration revision

The attribute **ConfRev** shall contain a count of the number of times that the configuration with regard to the **USVCB** has been changed. Changes that shall be counted are:

- any deletion of a member of the **DATA-SET**,
- any reordering of members of the **DATA-SET**,
- any change of a value of the **DataAttribute** of the **DATA-SET** whose functional constraint equals cf.

1 - any change of a value of an attribute of **USVCB** (functional constraint of attribute **USVCB**
2 equals **US**).

3 The counter shall be incremented when the configuration changes.

4 The initial value for **ConfRev** is outside the scope of this standard. The value of 0 shall be
5 reserved. A restart of the IED shall not reset the value.

6 NOTE Configuration changes of **DATA-SETs** due to processing of services are not allowed (see **DATA-SET** model).
7 Changes to be taken into account for the **ConfRev** are those made by local means like system configuration.

8 **17.3.1.8 SmpRate**

9 The attribute **SmpRate** shall specify the sample rate in units of samples per nominal period.

10 **17.3.1.9 OptFlds – optional fields to include in SV message**

11 The attribute **OptFlds** shall be the client-specified optional fields to be included in the **SV**
12 message issued by this **USVCB**. This attribute defines a subset of the optional header fields
13 that shall be included in the **SV** message:

- 14 - **RefrTm** (Refresh time, time of refresh activity)
- 15 - **SmpSynch** (Samples synchronized, samples are synchronized by clock signals), and
- 16 - **SmpRate** (sample rate from the instance of **USVCB**)

17 - **DatSet** (data set name)

Kommentar [KS156]: #18
8

18 **17.3.2 Unicast sampled value services**

19 **17.3.2.1 Overview**

20 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

21

22 **17.3.2.2 SendUSVMessage**

23 **17.3.2.2.1 SendUSVMessage parameter table**

24 The **SendUSVMessage** service shall be used by a **USVCB** to send sampled values from the
25 server to the client over a **TWO-PARTY-APPLICATION-ASSOCIATION**.

Parameter name
Request
USV message

26

27 **17.3.2.2.2 Request**

28 **USV message**

29 The parameter **USV message** shall specify the values of the members of the referenced
30 **DATA-SET** of the **USVCB** as specified in the abstract sampled value format definition
31 (see 16.4). The concrete format of the **USV message** shall be defined in the SCSM.

1 17.3.2.3 GetUSVCBValues

2 A client shall use the **GetUSVCBValues** service to retrieve attribute values of **USVCB** made
3 visible and thus accessible to the requesting client by the referenced **LLNO**.

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

6

Parameter name
Request
UsvCBReference
FunctionalConstraint
Response+
SvEnable
CBReserved
UnicastSampleValueID
DataSetReference
ConfigurationRevision
SampleRate
Response-
ServiceError

7

8 17.3.2.3.1 Request

9 17.3.2.3.1.1 UsvCBReference

10 The parameter **UsvCBReference** shall specify the **ObjectReference** of the **USVCB**.

11 The service parameter **UsvCBReference** shall be **LDName/LLNO.UsvCBNam**.

12 17.3.2.3.1.2 FunctionalConstraint

13 The parameter **FunctionalConstraint** (**FCCB**) shall contain the value of the functional con-
14 straint parameter to filter the respective instances of attributes of a **USVCB**.

15 The service parameter **FunctionalConstraint** (**FCCB**) shall be **US** (unicast sampled value
16 control).

17 17.3.2.3.2 Response+

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

19 17.3.2.3.2.1 SvEnable

20 The parameter **SvEnable** shall contain the value of the corresponding attribute **SvEna** of the
21 referenced **USVCB**.

22 17.3.2.3.2.2 CBReserved

23 The parameter **CBReserved** shall contain the value of the corresponding attribute **Resv** of
24 the referenced **USVCB**.

1 **17.3.2.3.2.3 UnicastSampleValueID**

2 The parameter **UnicastSampleValueID** shall contain the value of the corresponding attrib-
3 ute **UsvID** of the referenced **USVCB**.

4 **17.3.2.3.2.4 DataSetReference**

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

7 **17.3.2.3.2.5 ConfigurationRevision**

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

10 **17.3.2.3.2.6 SampleRate**

11 The parameter **SampleRate** shall contain the value of the corresponding attribute **SmpRate**
12 of the **USVCB**.

13 **17.3.2.3.3 Response-**

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

16 **17.3.2.4 SetUSVCBValues**

17 A client shall use the **SetUSVCBValues** service to set attribute values of **USVCB** made visi-
18 ble and thus accessible to the requesting client by the referenced **LLNO**.

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
UsvCBReference
FunctionalConstraint
SvEnable [0..1]
CBReserved [0..1]
UnicastSampleValueID [0..1]
DataSetReference [0..1]
SampleRate [0..1]
Response+
Response-
ServiceError

21

22 **17.3.2.4.1 Request**

23 **17.3.2.4.1.1 UsvCBReference**

24 The parameter **UsvCBReference** shall specify the **ObjectReference** of the **USVCB**.

25 The service parameter **UsvCBReference** shall be **LDName/LLNO.UsvCBNam**.

1 **17.3.2.4.1.2 FunctionalConstraint**

2 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional con-
3 straint parameter to filter the respective instances of attributes of a **USVCB**.

4 The service parameter **FunctionalConstraint (FCCB)** shall be **US** (unicast sampled value
5 control).

6 **17.3.2.4.1.3 SvEnable [0..1]**

7 The parameter **SvEnable** shall contain the value for the corresponding attribute **SvEna** of the
8 referenced **USVCB**.

9 **17.3.2.4.1.4 CBReserved**

10 The parameter **CBReserved** shall contain the value for the corresponding attribute **Resv** of
11 the referenced **USVCB**.

12 **17.3.2.4.1.5 UnicastSampleValueID [0..1]**

13 The parameter **UnicastSampleValueID** shall contain the value for the corresponding attrib-
14 ute **UsvID** of the referenced **USVCB**.

15 **17.3.2.4.1.6 DataSetReference [0..1]**

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

18 **17.3.2.4.1.7 SampleRate [0..1]**

19 The parameter **SampleRate** shall contain the value for the corresponding attribute **SmpRate**
20 of the **USVCB**.

21 **17.3.2.4.2 Response+**

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

23 **17.3.2.4.3 Response–**

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

26 This service shall return a failure if the service has been issued for any attribute of a **USVCB**
27 other than **SvEnable** while **USVCB** is enabled.

28 **17.4 Sampled value format**

29 The abstract sampled value format used for the sampled value message shall be as defined
30 in 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 .		

Kommentar [KS157]: #18
8

Kommentar [KS158]: #42

Kommentar [KS159]: #18
9

2

3 **17.4.1 MsvID or UsvID**

4 The parameter **MsvID** or **UsvID** shall contain the values of the attributes **MsvID** or **UsvID**
5 of the **MSVCB** or **USVCB** to be included in the sampled value message.

6 **17.4.2 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
10 **RefrTm** (**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 **17.4.3 DataSet**

14 The parameter **DatSet** (taken from the **MsvID** or **UsvID**) shall contain the **ObjectRefer-**
15 **ence** of the **DATA-SET** whose values of the members are transmitted in the message.

16 **17.4.4 Sample [1..n]**

17 The parameter **Sample** shall contain the value of a member of **DATA-SET** sampled at
18 a given time.

19 **17.4.5 SmpCnt**

20 The parameter **SmpCnt** shall contain the values of a counter, which is incremented each
21 time a new sample of the analogue value is taken. The sample values shall be kept in the
22 right order. If the counter is used to indicate time consistency of various sampled values, the
23 counter 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 **17.4.6 RefrTm**

2 The parameter **RefrTm** shall contain the time when the transmission buffer has been re-
3 freshed locally.

4 NOTE The semantic of the **RefrTm** is defined in the SCSM. This time may be used by the subscriber to check
5 the validity of the data.

6 **17.4.7 ConfRev**

7 The parameter **ConfRev** shall contain the value of the attribute **ConfRev** of the **MSVCB** or
8 **USVCB**.

9 **17.4.8 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

Kommentar [KS160]: #18
9

15 **17.4.9 SmpRate**

16 The parameter **SmpRate** shall contain the value of the attribute **SmpRate** of the **MSVCB** or
17 **USVCB**.

18

1 **18 CONTROL class model**

2 **18.1 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

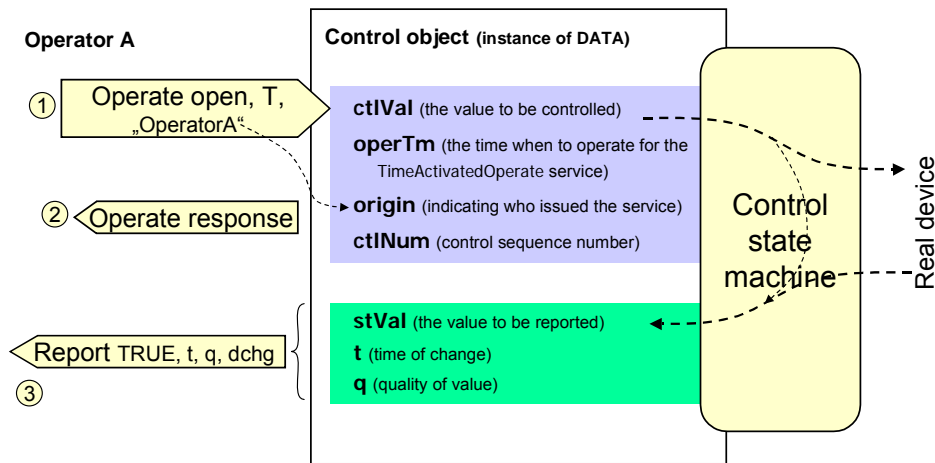
- 7 - specification of services;
- 8 - a behaviour described with state machines.

9 The control model defines the following services:

- 10 - **Select (Sel) / SelectWithValue (SelVal)**
- 11 - **Cancel**
- 12 - **Operate (Oper) / TimeActivatedOperate (TimOper)**
- 13 - **CommandTermination (CmdTerm)**

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 34.



IEC 424/03

18 **Figure 34 – 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:

- 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 shall be controlled by the parameter "check condition". The check condition may be specified by a service parameter (for example, synchrocheck). Besides the check condition specified by the service parameter, the control object 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).

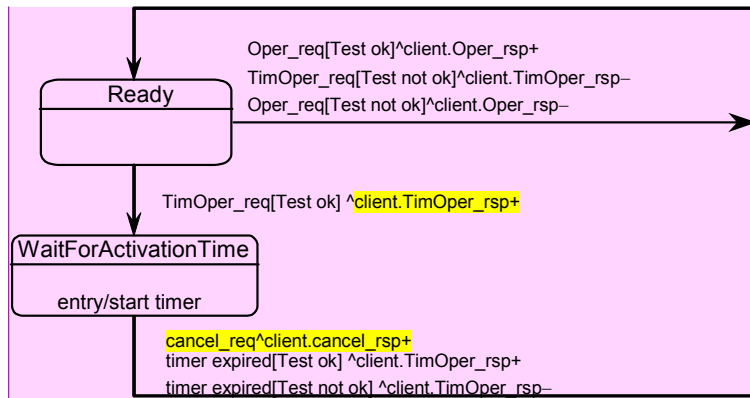
Kommentar [KS161]: Revised according to decision of editors' meeting in Zug.

11 **18.2 Control with normal security**

12 In the case of control with normal security there shall be no additional supervision of the status value by the control object. This means that for the negative case, if the status value did not change to the control value, the client will not get information about the failure from the control object.

16 **18.2.1 Direct control with normal security**

17 This model shall use the services **Operate** and **TimeActivatedOperate**.

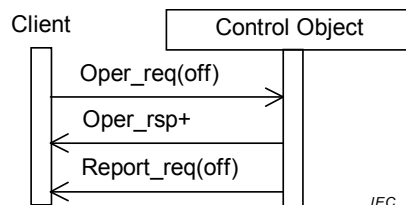


Kommentar [CB162]: TIS SUE #48;

IEC 425/03

20 **Figure 35 – State machine of direct control with normal security**

21 Direct control with normal security should be used for operations that act either on local **DATA** (for example, a LED test) or on **DATA** that influence external devices where a return information is not supervised (for example, switch on a heating).



IEC 426/03

25 **Figure 36 – 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.

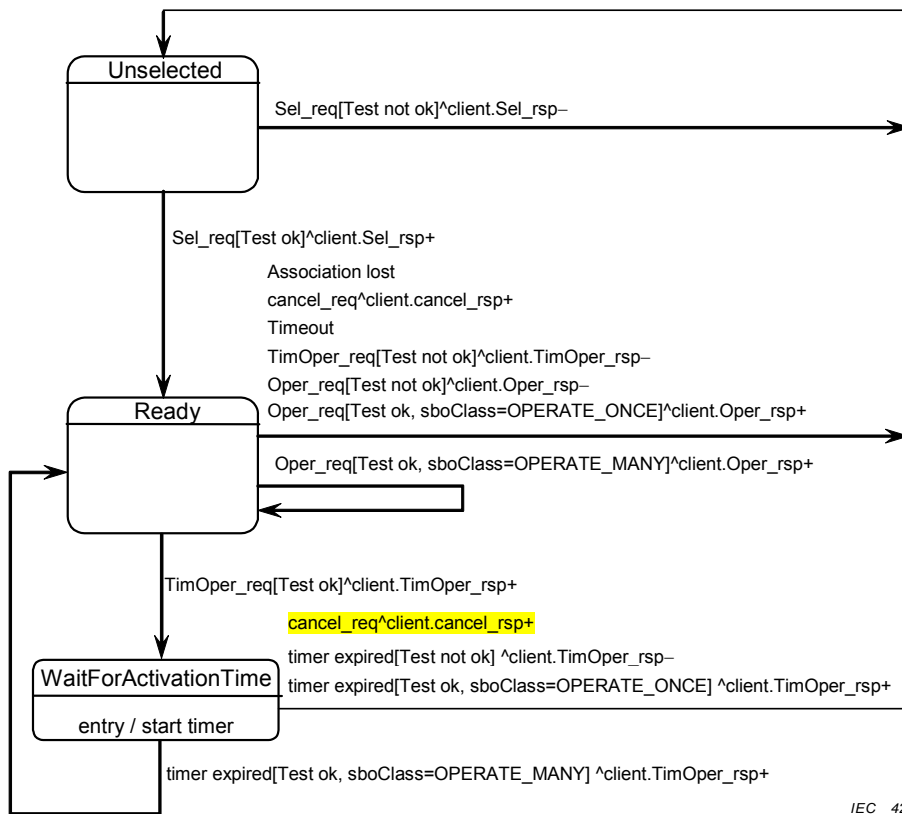
4 • If not successful, the control object shall issue a negative response to the requesting cli-
5 ent.

6 • If successful, the control object shall issue a positive response to the requesting client
7 and causes the requested action.

8 The new status may be reported by the **Report** service (see reporting model).

9 **18.2.2 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 se-
13 lected control object from other clients shall generate a negative response.



14

15 NOTE This state machine is compatible to the SBO control model defined in UCA™.2.

16 **Figure 37 – 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 appropriate access authority, that the control object is not currently selected by a different client, and that the device represented by the associated **LOGICAL-NODE** is operable and is not tagged so as to restrict operation.
- 3
- 4
- 5
- 6 – If the **Select** operation is not valid, the control object shall issue a negative response to the requesting client.
- 7
- 8 – If the **Select** operation is valid, the control object shall issue a positive response to the requesting client, shall change the state to ready and starts a deselect timer for either the interval defined by the **SelfTimeOut** attribute or, if unimplemented, some locally determined duration.
- 9
- 10
- 11
- 12 b) If the deselect timer expires before an **Operate** request on one or more of the other control components shall be requested by the selecting client, the control object shall change the state to unselected.
- 13
- 14
- 15 c) If an **Operate** request is received from the selecting client while the state is unselected for that client, the operation shall be denied.
- 16
- 17 d) On receipt of an **Operate** request, the control object shall check validation of the control execution.
- 18
- 19 – If not successful, the control object shall issue a negative response to the requesting client.
- 20
- 21 – If successful, the control object shall issue a positive response to the requesting client and shall cause the requested action by activating a binary output (or sending an equivalent signal on a process bus).
- 22
- 23

Kommentar [KS163]: #53
7; CHB ... to clarify

Kommentar [CB164]: TIS
SUE #331

24 18.3 Control with enhanced security

25 18.3.1 Introduction

26 In the case of control with enhanced security there shall be an additional supervision of the status value by the control object. Each command sequence shall be terminated by a **CommandTermination** service primitive.

27

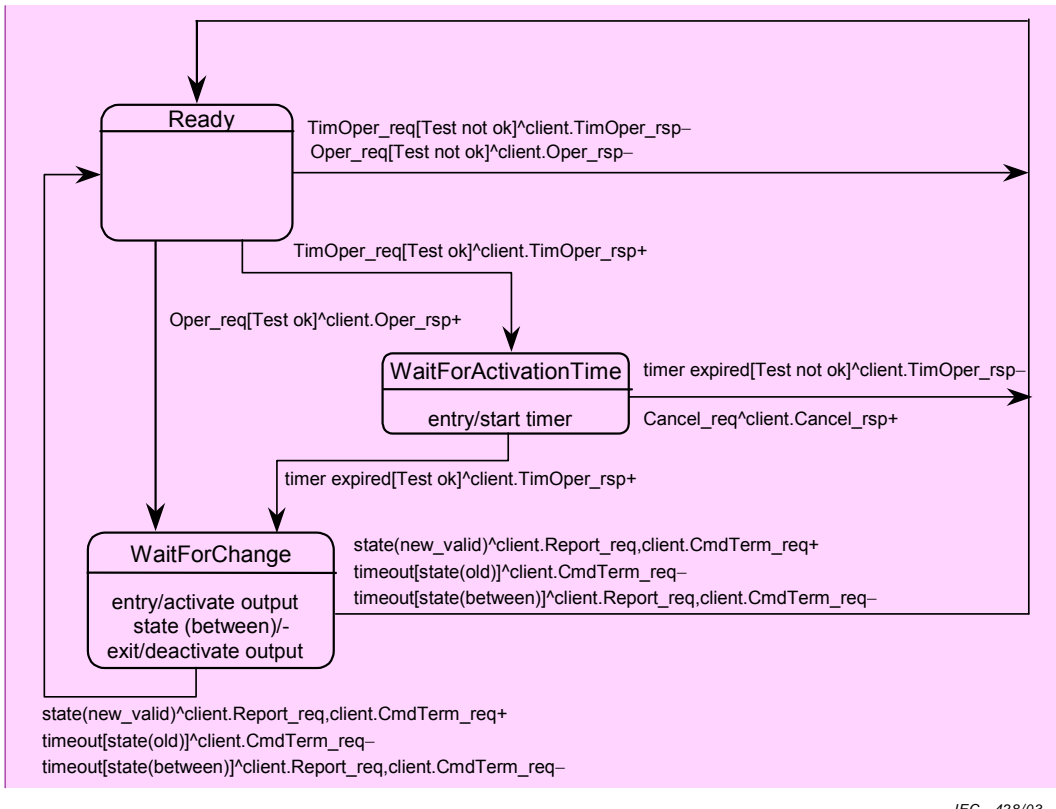
28

29 18.3.2 Direct control with enhanced security

30 This model shall use the services **Operate**, **TimeActivatedOperate**, and **CommandTermination**.

31

Kommentar [CB165]: Needs to be updated – take report away



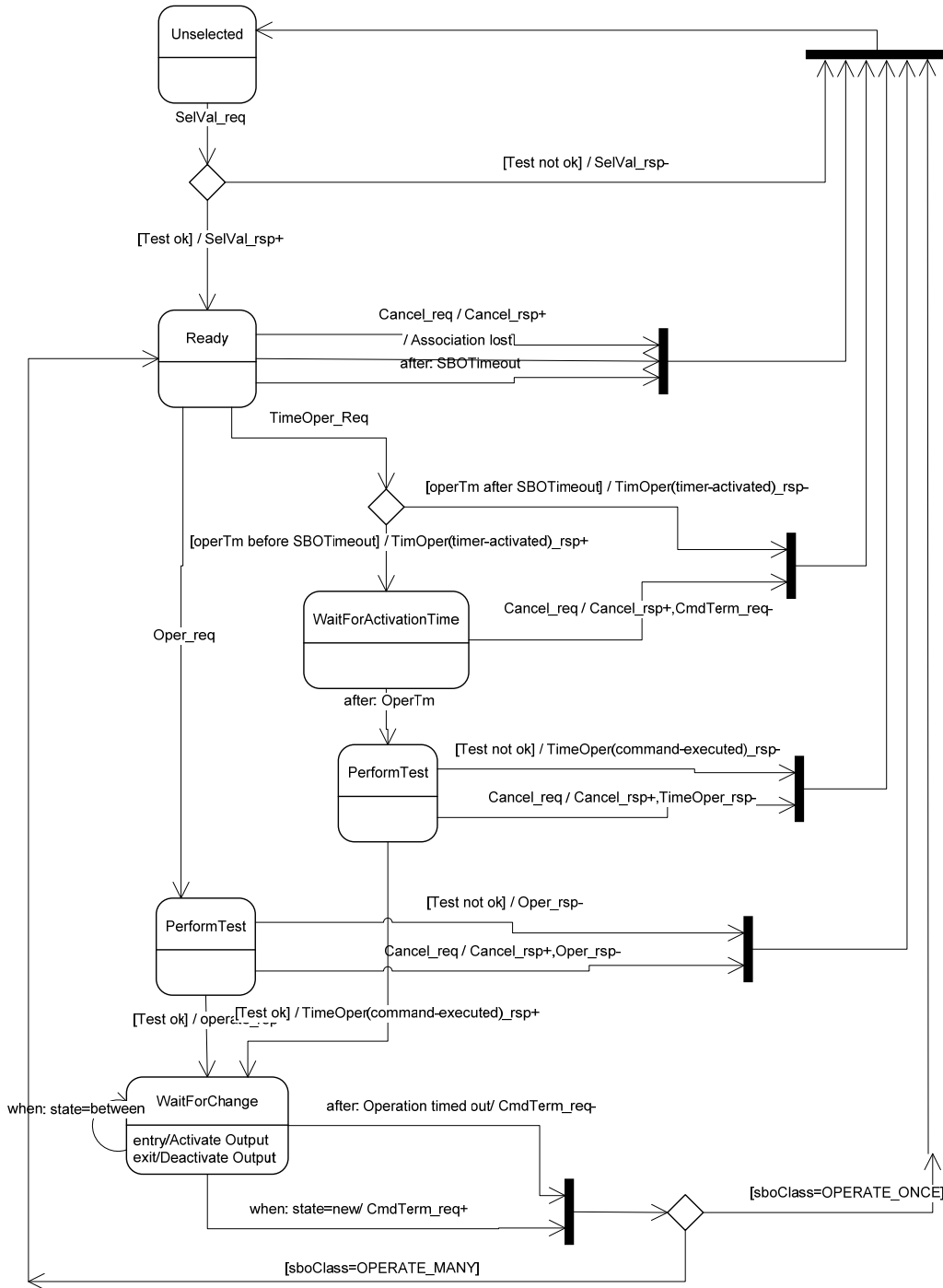
1
2

3 **Figure 38 – State machine of direct control with enhanced security**

4 NOTE – as a result of the state change, the status attribute of the control object may change and this change may
 5 be e.g. reported. In the following sequence diagrams, such a report is shown.

6 **18.3.3 SBO control with enhanced security**

7 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.
 8
 9
 10
 11



1
2
3
4
5

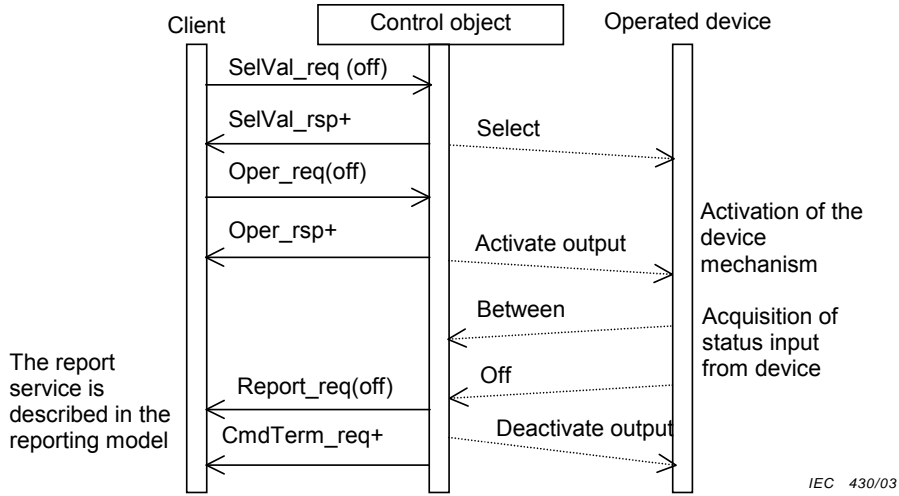
Figure 39 – State machine SBO control with enhanced security

Kommentar [CB166]: Check Update of statemachine according to TISSUE #46, 334

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.

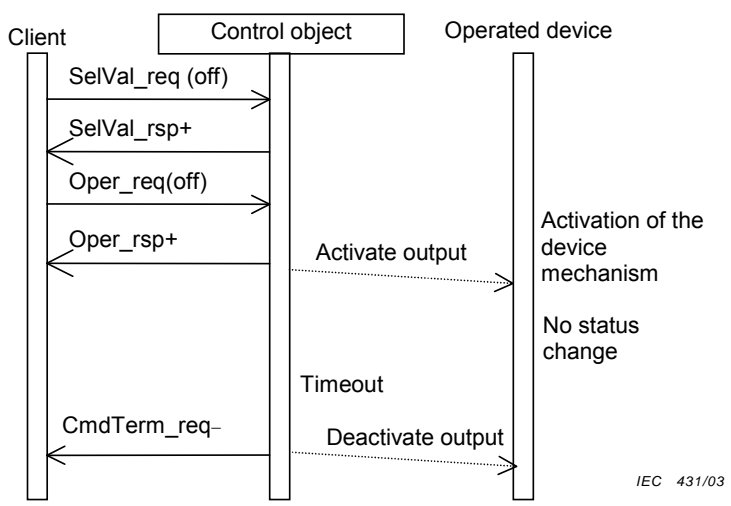
1 Control with enhanced security should be used for control procedures that cause an impor-
2 tant action outside the device containing the accessed control object.

Kommentar [CB167]: Check which sequence diagrams need to be updated (Fig 37); add a sequence diagram with the cancel after oper rsp+



3
4 **Figure 40 – Select before operate with enhanced security – positive case**

5 NOTE The dashed lines in Figure 40 and Figure 41 indicate that these "services" are local and not visible at the
6 communication level.



7
8 **Figure 41 – 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 dif-
- 13 ferent client, and that the device represented by the associated **LOGICAL-NODE** is oper-
- 14 able and is not tagged so as to restrict operation.
- 15 – If the **SelectWithValue** operation is not valid, the control object shall issue a nega-
- 16 tive response to the requesting client.

- 1 – If the **SelectWithValue** operation is valid, the control object shall issue a positive re-
 2 sponse to the requesting client, shall change the state to ready and starts a deselect
 3 timer for either the interval defined by the **sboTimeout** 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 con-
 6 trol components shall be requested by the selecting client, the control object shall change
 7 the 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 Wait-
 17 ForChange.
- 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
 20 report 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 ob-
 22 ject 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 per-
 27 formed 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
 31 Ready.
- 32 The last action shall be the command termination (**CmdTerm**) service.

Kommentar [KS168]: #53
 7; There needs to be a hint that
 this attribute is defined in 7-3!!

34 18.4 Time-activated operate

35 Time-activated control shall consist of a **TimeActivatedOperate** request and response. The
 36 response shall inform the requesting client whether the command was successful, and had
 37 caused a time activation process, or unsuccessful.

38 This shall be an extension of the control model. To use the time-activated operate capability
 39 the service Operate in the control model shall be replaced by the service **TimeActivated-**
 40 **Operate**.

41 NOTE The example below is shown with the sboClass direct-operate. The use of select before operate mode is
 42 also possible. In that case, the control object must be in the state Ready before the service TimeActivatedOperate
 43 is supported.

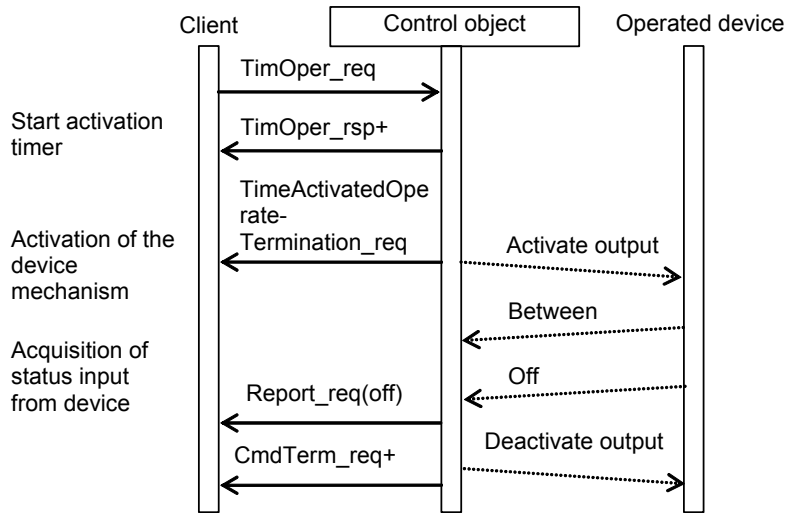


Figure 42 – Time-activated operate

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
6 client.

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
10 sent to the client.

11 c) All further information exchange shall be as described in the model for control with en-
12 hanced security.

13 **18.5 CONTROL class service definitions**

14 **18.5.1 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 18.5.2 Service parameter definition

2 The following service parameters shall be applied in the service definitions. Additional ser-
3 vice parameters for control may be defined with the respective common data classes.

Kommentar [CB169]: TIS
SUE #173

4 18.5.2.1 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"**.

Kommentar [KS170]: 1.1.1.
ext Clause removed accord-
ing to decision in Zug:
AddCause – additional cause
diagnosis

8 18.5.3 Service specification

9 18.5.3.1 General

10 With the common data classes, additional service parameters for control are defined. The
11 control services use these service parameters to communicate the specific values of the
12 service. All service parameters of a specific service need to be included in one service
13 request.

Kommentar [CB171]: TIS
SUE #173

14 **NOTE 1** The SCSM defines the subset of service parameters in the response service primitives. A communication
15 stack that allows the client to assign a response to the relating request may not support all the service parameters
16 that were also transmitted in the request.

Kommentar [KS172]: NOT
E 2 removed according to deci-
sion in Zug.

17 18.5.3.2 Select (Sel)

18 The **Select** service shall define the following service parameters.

Parameter name
Request
ControlObjectReference
Response+
ControlObjectReference
Response–
ControlObjectReference

19

20 NOTE The service parameters are defined in 17.5.2.

21

22 18.5.3.3 SelectWithValue (SelVal)

23 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

1

2 NOTE The service parameters are defined in 17.5.2.

3

4 **18.5.3.4 Cancel**

5 The **Cancel** service shall be used to abort a control operation. It is indicated in the state ma-
6 chines, in which states it is possible to cancel the control operation.

Kommentar [KS173]: Text changed according to decision of editors' meeting in Zug.

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

7

8 NOTE The service parameters are defined in 17.5.2.

9

1 **18.5.3.5 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

3

4 NOTE The service parameters are defined in 17.5.2.

5

6 **18.5.3.6 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

8

9 NOTE The service parameters are defined in 17.5.2.

10

1 **18.5.3.7 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
Response-
ControlObjectReference
Additional service parameters from common data class

3

4 NOTE The service parameters are defined in 17.5.2.

5 **18.5.3.8 TimeActivatedOperate_Termination**

6 tbd

7

Kommentar [KS174]: As discussed in Zug.

1 **19 Role based access supporting service model**

2 Editor's Note This clause requires some coordination with the new expected project on Role based access now
3 under way in TC 57: 57/914/NP

4 **19.1 General**

5 tbd

6 **19.2 Roles and List of users**

7 tbd

8 **19.3 Services**

9 **19.3.1 GetRole (role only, role plus users)**

10 tbd

11 **19.3.2 AddUserToRole**

12 tbd

13 **19.3.3 RemoveUserFromRole**

14 tbd

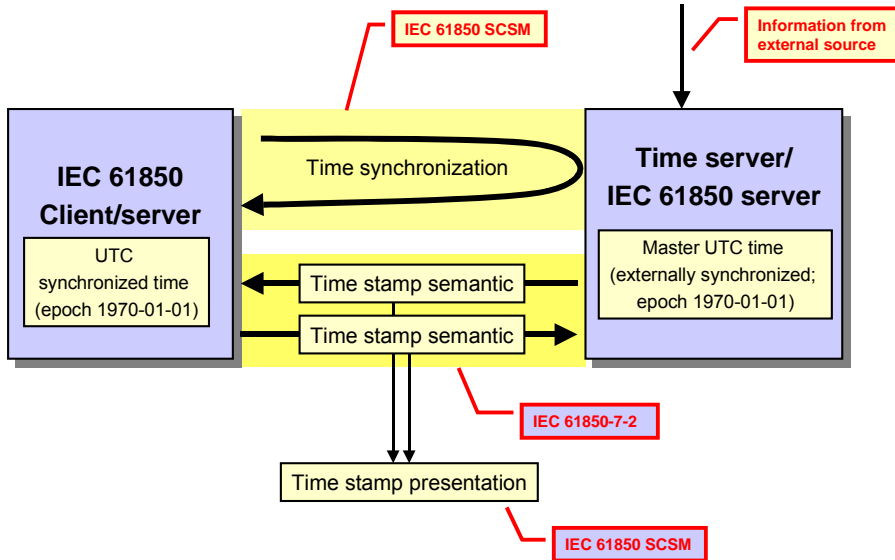
15

16

1 **20 Time and time-synchronization model**

2 **20.1 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 43.



6
7 **Figure 43 – 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 syn-
10 chronize 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 syn-
16 chronization 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 **20.2 External information**

21 External information required for the time and time synchronization model shall provide the
22 following.

- 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
27 server shall have the LeapSecondsKnown quality attribute set to true, otherwise set to
28 false.
- 29 b) **Epoch** (for example, GPS 6.1.1980).

1 21 Naming conventions

2 21.1 Class naming and class specializations

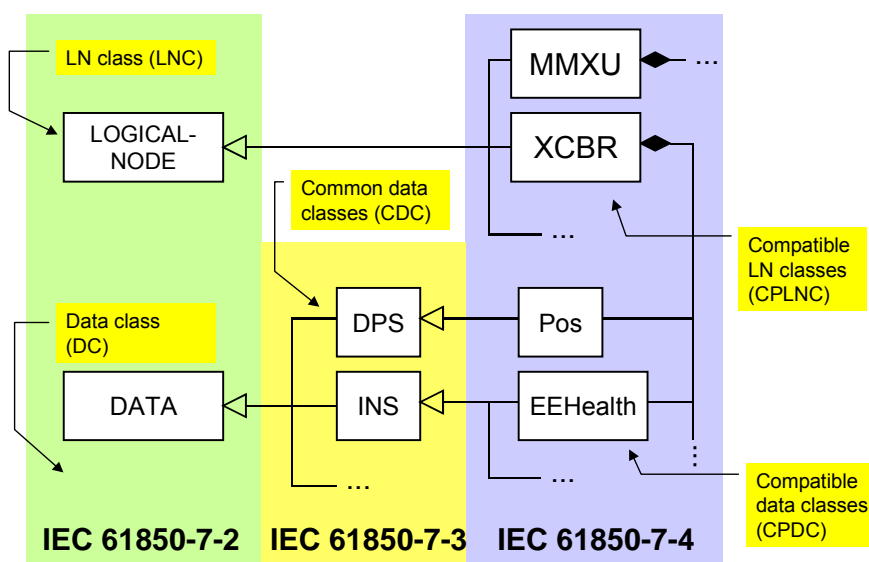
3 The classes for **DATA**, **common DATA**, **compatible DATA**, and **compatible LOGICAL-**
 4 **NODE** defined in IEC 61850-7-x make use of the following specializations:

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
 8 of 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 44 shows an overview of the specializations.



IEC 434/03

13

14

Figure 44 – 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 **21.2 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 37.

4 **Table 37 – 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.MsvCBName
USVCB	
UsvCBRef (multicast sampled value control block)	LDName/LLN0.UsvCBName

Kommentar [KS175]: Corrected see comment for DataAttributeReference

Kommentar [KS176]: Add ed "(" and ")". Note that the NumArrayElement cannot be mapped to the flattened name path of MMS variables. In MMS we have to use AlternatAccess ... which is more powerfull than what we define in 7-2. MMS allows also to select a range of element numbers.

Kommentar [KS177]: Due to the extensions of Array type and correction required anyway because the DataAttributeReference was not correct in Edition 1.

Kommentar [KS178]: Due to extensins in Log model.

1 Additionally, the following length definitions shall apply.

2 **References for accessing data:**

**LDName/LNName.
 DataName[.DataName[...]].DataAttributeName[.DAComponentName[...]]**

3 The inner square bracket “[...]” shall indicate further recursive definitions of nested data at-
 4 tribute components.

5 **References for accessing control blocks:**

LDName/LNName.CBName

Kommentar [KS179]: Add ed according to editors' meeting in Zug. To make the definition consistent.

6

7 **LDName** = up to 64 characters, application specific

Kommentar [KS180]: #23 6

8 **LNName** = [LN-Prefix] LN class name [LN-Instance-ID]
 9 LN-Prefix = m characters (application specific); it may start with any character
 10 LN class name = 4 characters (for example, compatible logical node name as defined
 11 in IEC 61850-7-4)
 12 LN-Instance-ID = n numeric characters (application specific)
 13 m+n ≤ 7 characters

Kommentar [KS181]: Ac- cording to editors' meeting in Zug, and according to MMS edition ISO 9506:2003.

14 **DataClassName** = up to 10 characters (as, for example, used in IEC 61850-7-4);
 15 no DataClassName shall end with a numeric character

16 **DataName** = DataClassName[Data-Instance-ID]

17 **Data-Instance-ID** = n numeric characters, optional; n shall be equal for all instances
 18 of the same data class]

Kommentar [KS182]: n shall be equal --- seems to be unclear!? There is no rule, e.g., if the numbers must be consecutive numbers etc.

19 **FCD** ≤ 61 characters including all separators "." (without the value of the FC)

20 **FCDA** ≤ 61 characters including all separators "." (without the value of the FC)

Kommentar [KS183]: Tis- sue #289

21 **CBName** = [CB-Prefix] CB class name [CB-Instance-ID]
 22 CB-Prefix = m characters (application specific)
 23 CB class name = 4 characters (as defined in this part of the standard)
 24 CB-Instance-ID = n numeric characters (application specific)
 25 m+n ≤ 7 characters

Kommentar [KS184]: #23 6 (old value of 29 plus 32)

Kommentar [KS185]: Mak e the definition consistent and complete

26 **FCCB** ≤ 22 characters including all separators "." (without the value of the FC)

Kommentar [KS186]: Mak e the definition consistent with CDV of part IEC 61850-6 (57/919). Are te values for m and n ok?

27 The characters allowed shall be:

28 VisibleString (FROM

29 ("A" | "a" | "B" | "b" | "C" | "c" | "D" | "d" | "E" | "e" | "F" | "f" |
 30 "G" | "g" | "H" | "h" | "I" | "i" | "J" | "j" | "K" | "k" | "L" | "l" |
 31 "M" | "m" | "N" | "n" | "O" | "o" | "P" | "p" | "Q" | "q" | "R" | "r" |
 32 "S" | "s" | "T" | "t" | "U" | "u" | "V" | "v" | "W" | "w" | "X" | "x" |
 33 "Y" | "y" | "Z" | "z" | "_" | "0" | "1" | "2" | "3" | "4" | "5" | "6" |
 34 "7" | "8" | "9"))

Kommentar [KS187]: Len gth needs to be defined. Can not be more than 11 (LN) + 11 (CB) ?

1 EXAMPLE Figure 45 shows examples of object names and object references. The example at the top
 2 (first five lines) can be just five class definitions (not yet instantiated) or five instances of the classes
 3 E1.QA5/XCBR.Pos.ctlVal, ...stVal, ...q, ...t, ...ctlModel. The object references in this case do not indicate if ob-
 4 ject references refer to classes or instances. The context in which these references are used has to provide suffi-
 5 cient information to know what is meant (just class or instance).

6 The other examples refer to instances only.

7 NOTE 1 The LD name E1.QA5 and its structure are outside the scope of IEC 61850. The functional constraint
 8 (FC) is not shown in the object reference. The FC information may be mapped into the **ObjectReference** in an
 9 SCSM; IEC 61850-8-1 maps the FC between LN and Data.

10 NOTE 2 The type of the **ObjectReference** is VISIBLESTRING129; this is different to the VisibleString (as de-
 11 fined above). The visibleString does not provide the separators ".", "/" and "()".

12

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 reference							

IEC 435/03

13

14 **Figure 45 – Object names and object reference**

15 **21.3 Scope**

16 **Server specific scope** (instances are defined outside of all LDs but in the server) shall be
 17 defined using the "/" and up to **64 characters** to the right.

18 EXAMPLE /ABC.xyz

19 **Logical device specific scope** (instances are defined inside a specific LD) shall be defined
 20 as up to **64 characters**, then "/" followed by up to **64 characters** to the right.

21 EXAMPLE Atlanta_110/XCBR.Pos

22 **TPAA specific scope** (instances are defined inside a specific TPAA) shall be defined using
 23 "@", then "/" followed by up to **64 characters** to the right.

24 EXAMPLE @/DataSet5 (for non-persistent DATA-SETs).

- 1 NOTE 1 The SCSMs may map the Reference to a flat numerical index or to a character string that is derived
2 from the definition above. These character strings may comprise additional elements such as the functional con-
3 straint (FC).
- 4 NOTE 2 IEC 61850-6 gives additional definitions on how the application-specific character strings for logical de-
5 vices can be built.

1 **22 File transfer**

2 **22.1 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 38.

10 **Table 38 – 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		

Kommentar [KS188]: #40
3

11

12 **22.1.1 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 **22.1.2 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 **22.1.3 LastModified**

21 The attribute **LastModified** shall be the time when the file was last modified.

1 **22.2 File services**2 **22.2.1 GetFile**3 **22.2.1.1 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 **22.2.1.2 Request**8 **FileName**

9 The parameter **FileName** shall specify the name of the file being transferred.

10 **22.2.1.3 Response+**

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

Kommentar [KS189]: #40
4

16 **22.2.1.4 Response–**

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

19 **22.2.2 SetFile**20 **22.2.2.1 SetFile parameter**

21 The **SetFile** service shall be used by a client to transfer the contents of a file from the client
22 to the server.

Parameter name
Request
FileName
File-Data
Response+
Response–
ServiceError

23

1 **22.2.2.2 Request**

2 **22.2.2.2.1 FileName**

3 The parameter **FileName** shall specify the name of the file being transferred.

4 **22.2.2.2.2 File-Data**

5 The parameter **File-Data** shall contain the data transferred; the type of file-data is **OCTET**
6 **STRING**.

Kommentar [KS190]: #40
5

7 **22.2.2.3 Response+**

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

Kommentar [KS191]: #40
5

10 **22.2.2.4 Response-**

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

13 **22.2.3 DeleteFile**

14 **22.2.3.1 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 **22.2.3.2 Request**

18 **FileName**

19 The parameter **FileName** shall specify the name of the file being deleted.

20 **22.2.3.3 Response+**

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

Kommentar [KS192]: #40
5

23 **22.2.3.4 Response-**

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

1 22.2.4 GetFileAttributeValues

2 22.2.4.1 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

Kommentar [KS193]: #40
7

5

6 22.2.4.2 Request

7 FileName

8 The parameter **FileName** shall, when present, specify the name of the file whose attributes
9 are requested to be returned to the client.

10 22.2.4.3 Response+

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

13 22.2.4.3.1 FileName

14 The parameter **FileName** shall provide the name of the file whose attributes are returned.

15 22.2.4.3.2 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 22.2.4.3.3 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.

Kommentar [KS194]: 407

21 22.2.4.4 Response-

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

24

25

26

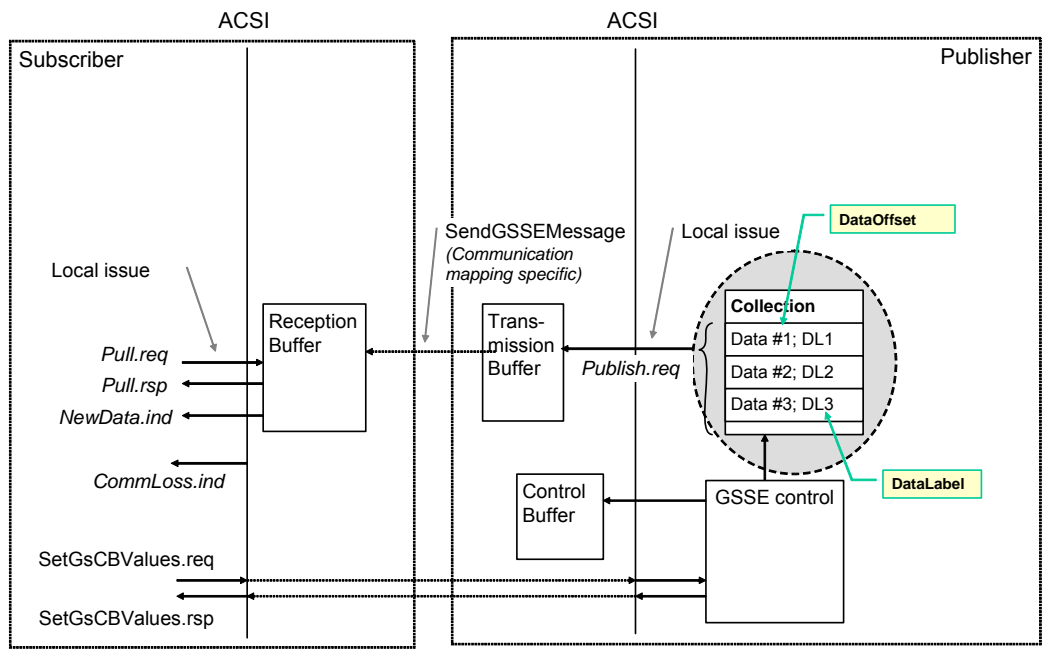
Annex A
(informative)

Generic substation state event (GSSE) control block (GsCB)

Kommentar [KS195]: Moved to annex according to decision of WG10

A.1 GsCB class definition

The specifics for the **GsCB** model (compared to the **GoCB** model) are depicted in the shadowed area in Figure 1.



IEC 422/03

Figure 1 - GsCB model

Kommentar [KS196]: #52
3

8
9

10

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 1.

1 **Table 1 – 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			

Kommentar [CB197]: TIS SUE #41

Kommentar [CB198]: TIS SUE #53

2

3 **A.2 Generic substation state event (GSSE) control block class attributes**

4 **A.2.1 GsCBName – GSSE control name**

5 The attribute **GsCBName** shall unambiguously identify a **GsCB** within the scope of a **LLNO**.

6 **A.2.2 GsCBRef – GSSE control reference**

7 The attribute **GsCBRef** shall be the unique path-name of a **GsCB** within a **LLNO**.

8 The ObjectReference **GsCBRef** shall be:

LDName/LLNO.GsCBName

9 **A.2.3 GsEna – GSSE enable**

10 The attribute **GsEna** (if set to TRUE) shall indicate that **GsCB** is currently enabled to send
 11 values of the **GsCB**. If set to FALSE the **GsCB** shall stop sending **GSSE** messages.

12 While being TRUE (**GsCB** enabled), no changes of attribute values of the **GsCB** other than
 13 disabling shall be allowed.

14 If the **TWO-PARTY-APPLICATION-ASSOCIATION** to the client that has enabled the **GsCB**
 15 is lost, the instance of **GsCB** shall set the attribute to FALSE.

16 **A.2.4 GsID – GSSE identification**

17 The attribute **GsID** shall be a user definable unique identification of the context of the GSSE
 18 message.

19 NOTE 1 The context of the GSSE message is defined by the values configured in the **GsCB**.

20 NOTE 2 Depending upon the SCSM and actual implementation, it may not be possible to uniquely identify the
 21 GSSE control through the control reference. Therefore, a standardized control attribute must be provided to allow
 22 the system configuration process to be able to uniquely identify the control within the scope of the substation.

1 **A.2.5 DataLabel [1..n]**

2 The attribute **DataLabel** of visible strings shall contain a reference for each entry used within
3 the attribute **LastSentData**. A NULL value shall indicate that particular **LastSentData** data
4 entry is not in use. The DEFAULT value is a local issue.

Kommentar [KS199]: Re-
moved one "that"

5 The visible string shall hold the value of the **ObjectReference** if the corresponding element
6 is being sent. Otherwise the value of the **ObjectReference** shall be NULL. The DEFAULT
7 value shall be **GsCBName**.

8 NOTE The attribute **DataLabel** allows a user to assign a system unique identifier for the application that is issu-
9 ing the **GSSE**.

10 **A.2.6 LSentData [1..n] – last sent data values**

11 The attribute **LSentData** shall represent the data values that have been sent with the last
12 **GSSE** message.

13 The maximum for the number of data values shall be at least 24; i.e. the attribute **LSentData**
14 shall be capable of holding at least 24 double-bit status values.

15 NOTE The maximum number of data values may be constrained by the SCSM and local means.

16 **A.2.7 DstAddress**

17 The attribute **DstAddress** shall be the SCSM specific addressing information like media ac-
18 ccess address, priority, and other information.

Kommentar [CB200]: TIS
SUE #53

19 **A.3 GSSE service definitions**

20 **A.3.1 Overview**

21 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 mes- sage
GetGSSEDataOffset	Retrieve the position of the specific value associated with the GSSE mes- sage of a DataLabel
GetGsCBValues	Retrieve the attributes of a GsCB
SetGsCBValues	Write the attributes of a GsCB

Kommentar [CB201]: TIS
SUE #417

22

23 **A.3.2 SendGSSEMessage**

24 **A.3.2.1 SendGSSEMessage parameter table**

25 The **SendGSSEMessage** service shall be used by a **GsCB** to send a **GSSE** message over a
26 **MULTICAST-APPLICATION-ASSOCIATION**.

Parameter name
Request
GSSE message

27

1 **A.3.2.2 Request**2 **A.3.2.2.1 GSSE message**

3 The parameter **GSSE message** shall specify the **GSSE** message as defined in 15.3.4 of the
4 given **GsCB**.

5 **A.3.2.3 GetGsReference**6 **A.3.2.3.1 GetGsReference parameter table**

7 A client shall use the **GetGsReference** service to retrieve the **DataLabels** of specific mem-
8 bers of the **Collection** of the referenced **GsCB**.

Parameter name
Request
GsCBReference
DataOffset [1..n]
Response+
GsCBReference
DataLabel [1..n]
Response-
ServiceError

9

10 **A.3.2.3.1.1 Request**11 **GsCBReference**

12 The parameter **GsCBReference** shall identify the attribute **GsCBRef** of the **GsCB** for which
13 **DataLabels** are being requested.

14 **DataOffset [1..n]**

15 The parameter **DataOffset** shall contain a number identifying a member of the **Collection**.

16 **A.3.2.3.1.2 Response+**17 **GsCBReference**

18 The parameter **GsCBReference** shall contain the parameter that identifies the attribute
19 **GoCBRef** of the **GsCB** for which **DataLabels** are returned.

20 **DataLabel [1..n]**

21 The parameter **DataLabel** shall contain the **DataLabel** requested for the **DataOffset** of the
22 **Collection**. A value of NULL shall indicate that no member is defined for the member being
23 requested with the respective **DataOffset**.

24 **A.3.2.3.1.3 Response-**

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

1 **A.3.2.4 GetGSSEDataOffset**

2 **A.3.2.4.1 GetGSSEDataOffset parameter table**

3 A client shall use the **GetGSSEDataOffset** service to retrieve the data position of a selected
4 data in the **Collection** associated with a **GsCB**.

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

7

Parameter name
Request
GsCBReference
DataLabel [1..n]
Response+
GsCBReference
DataOffset [1..n]
Response-
ServiceError

8

9 **A.3.2.4.1.1 Request**

10 **GsCBReference**

11 The parameter **GsCBReference** shall identify the attribute **GsCBRef** of the **GsCB** for which
12 **MemberOffset** are being requested.

13 **DataLabel [1..n]**

14 The parameter **DataLabel** shall contain the **DataLabel** for which the **DataOffset** of the **Col-**
15 **lection** is requested.

16 **A.3.2.4.1.2 Response+**

17 **GsCBReference**

18 The parameter **GsCBReference** shall contain the parameter that identifies the attribute
19 **GoCBRef** of the **GsCB** for which **DataLabels** are returned.

20 **DataOffset [1..n]**

21 The parameter **DataOffset** shall contain a number identifying a member of the **Collection**.
22 A value of NULL shall indicate that no **DataOffset** is defined for the member being requested
23 with the respective **DataLabel**.

24 **A.3.2.4.1.3 Response-**

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

27 **A.3.2.4.2 GetGsCBValues**

28 A client shall use the **GetGsCBValues** service to retrieve attribute values of **GsCB** made visi-
29 ble 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 the
2 view concept).

3

Parameter name
Request
GsCBReference
FunctionalConstraint
Response+
GsEnable
GSSEID
DataLabel [1..n]
LastSentData [1..n]
Response-
ServiceError

4

5 **A.3.2.4.2.1 Request**

6 **GsCBReference**

7 The parameter **GsCBReference** shall specify the **ObjectReference** of the **GsCB**.

8 The service parameter **GsCBReference** shall be **LDName/LLN0.GsCBName**.

9 **FunctionalConstraint**

10 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional con-
11 straint parameter to filter the respective instances of attributes of a **GsCB**.

12 The service parameter **FunctionalConstraint (FCCB)** shall be **GS** (gsse control).

13 **A.3.2.4.2.2 Response+**

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

15 **GsEnable**

16 The parameter **GsEnable** shall contain the value of the corresponding attribute **GsEna** of the
17 referenced **GsCB**.

18 **GSSEID**

19 The parameter **GSSEID** shall contain the value of the corresponding attribute **GsID** of the
20 referenced **GsCB**.

21 **DataLabel [1..n]**

22 The parameter **DataLabel** shall contain the **DataLabel** of the **Collection**.

23 **LastSentData [1..n]**

24 The parameter **LastSentData** shall contain the value of the attribute **LSentData** of the
25 **GsCB**.

1 **A.3.2.4.2.3 Response-**

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

4 **A.3.2.4.3 SetGsCBValues**

5 A client shall use the **SetGsCBValues** service to set attribute values of **GsCB** made visible
6 and thus accessible to the requesting client by the referenced **LLNO**.

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

9

Parameter name
Request
GsCBReference
FunctionalConstraint
GsEnable [0..1]
GSSEID [0..1]
Response+
Response-
ServiceError

10

11 **A.3.2.4.3.1 Request**

12 **GsCBReference**

13 The parameter **GsCBReference** shall specify the ObjectReference of the **GsCB**.

14 The service parameter **GsCBReference** shall be **LDName/LLNO.GsCBName**.

15 **FunctionalConstraint**

16 The parameter **FunctionalConstraint (FCCB)** shall contain the value of the functional con-
17 straint parameter to filter the respective instances of attributes of a **GsCB**.

18 The service parameter **FunctionalConstraint (FCCB)** shall be **GS** (gsse control).

19 **GsEnable [0..1]**

20 The parameter **GsEnable** shall contain the value for the corresponding attribute **GsEna** of the
21 referenced **GsCB**.

22 **GSSEID [0..1]**

23 The parameter **GSSEID** shall contain the value for the corresponding attribute **GsID** of the
24 referenced **GsCB**.

25 **A.3.2.4.3.2 Response+**

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

1 **A.3.2.4.3.3 Response–**

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

4 This service shall return a failure if the service has been issued for any attribute of a **GsCB**
5 other than **GsEnable** while **GsCB** is enabled.

6 **A.3.2.5 Generic substation state event (GSSE) message**

7 The abstract **GSSE** message format shall specify the information to be included in the **GSSE**
8 message. The structure of the **GSSE** message shall be as specified in Table 2.

9 A **GSSE** message shall at least be sent each time when a value from one or more of the
10 **LSentData** change (for example, a change of status value is detected).

11 **Table 2 – 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)

12

13 **GsID – application identifier**

14 The parameter **GsID** shall contain the value of the attribute GsID of the GsCB.

15 **T – time stamp**

16 The parameter **T** shall contain the time at which the **StNum** attribute was incremented.

17 **SqNum – sequence number**

18 The parameter **SqNum** shall contain the counter that shall increment each time a GSSE mes-
19 s- age has been sent.

20 The initial value for **STNum** shall be 1. The value of 0 shall be reserved.

21 **StNum – state number**

22 The parameter **StNum** shall contain a counter that increments each time a GSSE message
23 has been sent and a value change has been detected within the data values of **LSentData**.

24 The initial value for **StNum** shall be 1. The value of 0 shall be reserved.

1 **Test – test**

2 The parameter **Test** shall indicate with the value of TRUE that the values of the message
3 shall not be used for operational purposes.

4 **PhsID – phase identification**

5 The parameter PhsID shall indicate faulted phases.

6 **GSSEData [1..n]**

7 The parameter **GSSEData** shall be a status value of 4 values CODED ENUM. The
8 defined values are invalid or transient (0) | false or closed (1) | true or open (2) | invalid (3).

9 The size of the array [1..n] is determined by the size of the **LSentData** attribute of the asso-
10 ciated **GsCB**.

11

12

13

Annex B (normative)

ACSI conformance statement

B.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**.

B.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 **B.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	

Kommentar [KS202]: #33
7

Kommentar [KS203]: #33
7

		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				

Kommentar [KS204]: #46
 1. The numbers M12-1 and M12-2 have been deleted. The other numbers will not change! The gap is acceptable.

Kommentar [KS205]: Comment CHB, Time and File Transfer independent of SVC

1

2 **B.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

3

Annex C
(informative)
SCL conformance

Editor's Note This Annex has been copied from IEC 61850-8-1. It is a mix of service model that fits into 7-2 and mapping specific information that fits into 8-1. This annex needs a major revision. To be done after the meeting in Fredericia.

C.1 General

Defines several degrees of conformance for which implementations may declare support of the substation configuration language.

Table C.1 – SCL conformance degrees

	SCL conformance	Client-CR			Server-CR		
		Base	F/S	Value/range	Base	F/S	Value/range
SCL.1	SCL file for implementation available (offline)				m	m	
SCL.2	SCL file available from implementation online	o	o		o	o	
SCL.3	SCL implementation reconfiguration supported online	o	o		o	o	

Conformance to SCL.1 is mandatory. The availability of an electronic instance of an SCL file shall be provided. This file shall be able to be parsed.

Implementations claiming conformance to SCL.2 or SCL.3 may support the ACSI services defined in Table D.2.

Table C.2 – Supported ACSI services for SCL.2 and SCL.3

	SCL Conformance	Client-CR			Server-CR		
		Base	F/S	Value/range	Base	F/S	Value/range
ACSI services							
	GetFileAttributeValues	o	o		o	m	
	GetFile	o	c2		o	c2	
	SetFile	o	c3		o	c3	
	DeleteFile	o	o		o	c3	
	GetDataValues	o	c1		o	c1	
	SetDataValues	o	c2		o	c2	
	SCL Control Block	i	i		o	c3	
	SCL File Structure	i	m		i	m	
	Remote Creation of SCL File	i	o		i	o	
c1 Shall be 'm' if support for SCL.2 or SCL.3 is declared. c2 Shall be 'm' if support for SCL.2 is declared. c3 Shall be 'm' if support for SCL.3 is declared.							

1 The additional MMS services of Table D.3 shall be supported if support for SCL.2 or SCL.3 is
 2 declared.

3 **Table C.3 – Additional MMS services for SCL.2 and SCL.3**

	SCL conformance	Client-CR			Server-CR		
		Base	F/S	Value/range	Base	F/S	Value/range
MMS services							
	GetCapabilityList	o	i		o	c3	
	GetDomainAttributes	o	o		o	m	
	LoadDomain	o	c1		o	c1	
	StoreDomain	o	c2		o	c2	
c1 Shall be 'm' if support for SCL.3 is declared. c2 Shall be 'm' if support for remote creation of a SCL is declared. c3 An implementation that claims implementation of the GetCapabilityList service shall return the list of service capabilities that contain the list of services defined in the SCL file. Note that more capabilities are allowed to be returned than is defined in the SCL. If no capabilities are defined in the SCL file, an implementation implementing GetCapabilityList may return a NULL list of capabilities.							

Kommentar [hsf206]:
Tissue 119

4

5 **C.2 SCL control block**

6 The SCL control block shall have a functional constraint of "SC". This control block shall occur
 7 in LLN0 only.

8 The SCL control block shall be a structured MMS TypeDefinition that contains the ordered
 9 NamedComponents defined in Table D.4.

10 **Table C.4 – Definition of SCL control block**

IEC 61850-8-1 component name	MMS TypeDescription	r/w	m/o	Comments
validate	VISIBLE-STRING size of 64 octets	r w	m o	Shall be 'm' if support for remote activation of a SCL is declared.
valState	Unsigned Integer - 8 bits	r	m	(0) – NOT-VALIDATED (1) – VALIDATION-ERROR (2) – VALIDATED (3) – VALIDATION-IN-PROGRESS (4) – NOT-SUPPORTED (5) – VALIDATE-FILE-PRESENT
activate	VISIBLE-STRING size of 64 octets	r w	m o	Shall be 'm' if support for remote activation of a SCL is declared.

11

12 **C.2.1 validate**

13 The validate attribute of the SCL control block shall contain the value of the last filename, including
 14 file directory path, that was requested to be validated.

15 If remote activation of an SCL file is declared to be supported, then a V-Put to this attribute
 16 shall cause a local process to start a validation process on the filename whose name was
 17 written. A V-Put to this attribute when valState = VALIDATION-IN-PROGRESS shall fail.

18 A successful validation, when the validated filename does not previously exist, shall cause
 19 the file to be moved from the "notvalidated" directory to the "validated" directory. If the file-

1 name previously exists, the valState shall be set to VALIDATE-FILE-PRESENT file shall not
2 be moved.

3 **C.2.2 valState**

4 This attribute shall contain the status of the validation phase that resulted from a V-Put to the
5 validate attribute.

6 **C.2.3 activate**

7 The activate attribute of the SCL control block shall contain the value of the last successfully
8 activated SCL file.

9 If remote activation of an SCL file is declared to be supported, then a V-Put to this attribute
10 shall cause a local process to:

- 11 – cause all associations to be terminated via a MMS abort service.
- 12 – shall not allow associations to be established prior to reconfiguration.
- 13 – shall reconfigure the LogicalDevice or entire device depending upon the scope of activa-
14 tion.
- 15 – shall allow associations to be re-established after reconfiguration is completed.

16 **C.3 SCL support within devices**

17 This Clause standardizes the file directory structure for the storage and retrieval of SCL files.
18 Additionally, this Clause defines a model, and control block, through which SCL files can be
19 written to a device, validated on the device, and then activated as the current configuration of
20 that device. It is the intent of this Clause to specify mechanisms through which the entire de-
21 vice (e.g. all LogicalDevices) or specific LogicalDevices may be reconfigured.

22 Support for SCL mandates that 3 domain attributes be supported. These attributes are
23 named:

- 24 – SCL-Version: the value of this attribute shall indicate the version of the last SCL file that
25 was successfully activated. If the version is unknown, the value shall be “unknown”.
- 26 – SCL-Revision: the value of this attribute shall indicate the revision of the last SCL file that
27 was successfully activated. If the revision is unknown, the value shall be “unknown”.
- 28 – SCL-FileName: the value of this attribute shall indicate the name of the SCL that was
29 successfully activated. If the filename is unknown, the value shall be “unknown”.

30 Implementations declaring support for SCL.2 shall support these domain attributes, the SCL
31 control block, and shall support the MMS GetDomainAttributes service.

32 **C.3.1 File directory structure for SCL files**

33 The file directory structure for SCL files shall augment the structure specified in 23.1. The
34 root directory name shall be “SCL”. Under the root shall be two subdirectories: “notvalidated”
35 and “validated”. SCL files may be placed (e.g. SetFile service) into the “notvalidated” direc-
36 tory only. The transfer of the SCL file, from “notvalidated” to “validated” shall occur when a
37 SCL file is validated locally. This validation phase shall be accomplished through the setting
38 of the “validate” attribute of the SCL control block. A locally created SCL file shall be created
39 in the “validated” directory only.

40 **C.3.2 Retrieval of an SCL file from an implementation**

41 Retrieval of an SCL file shall be per the GetFile ACSI service. This service shall be executed
42 on the files in the directory structure specified in 23.2.1.

1 **C.3.3 Remote creation of a SCL file**

2 The ability of a client to create a SCL file remotely can be accomplished through the MMS
3 StoreDomainContent service.

4 If the StoreDomainContent service is executed on LogicalDevice LD0 (as defined in
5 IEC 61850-7-1), then the generated SCL file shall contain the contents for the entire device.

6 If the StoreDomainContent service is executed on a LogicalDevice, other than LD0, then the
7 contents of the SCL file shall contain at least the SCL information pertaining to that Logical-
8 Device.

9 The file directory, into which the generated SCL file is to be placed, shall have the root of
10 "SCL\validated". If the root is not present, or if the file already exists, the StoreDomainCon-
11 tent service shall fail and no SCL file shall be generated.

12 **C.3.4 Local validation of an SCL file**

13 The methodology for validating an SCL file is a local issue.

14

15

16

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

Kommentar [KS207]: #44
9

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					