ELCOM-90
Local Conventions

ELCOM Working Group
Convener Birger Stene

May 2008

TECHNICAL REPORT

SINTEF Energy Research

Address: NO-7465 Trondheim, NORWAY
Reception: Sem Sælands vei 11
Telephone: +47 73 59 72 00
Telefax: +47 73 59 72 50

www.energy.sintef.no

Enterprise No.:
NO 939 350 675 MVA

SUBJECT/TASK (title)

ELCOM-90 Local Conventions

CONTRIBUTOR(S)

Nils Eggen, Powel ASA, Lars Rindal, Siemens as, Tormod Lund, ABB, Anne Grethe Bolsø, ABB, Birger Stene, SINTEF Energiforskning AS

CLIENTS(S)

Statnett SF

TR NO.	DATE	CLIENT'S REF.	PROJECT NO.
TR A3933.01	2008-05-01	Anders Larsen	12X513

ELECTRONIC FILE CODE	RESPONSIBLE (NAME, SIGN.)	CLASSIFICATION
	Birger Stene	Unrestricted

ISBN N0.	REPORT TYPE	RESEARCH DIRECTOR (NAME, SIGN)	COPIES	PAGES
82-594-0395-1		Petter Støa	5	68

DIVISION	LOCATION	LOCAL FAX
Energy Systems	Sem Sælandsv. 11, 7465 Trondheim	+47 73 59 72 50

RESULT (summary)

This document is one of a series of technical reports which form the complete ELCOM-90 documentation. This is version .01 of the report with minor changes regarding responsible people and references. Future updates and new versions will NOT be published for this reason. New versions will only be submitted when technical changes are made.
Please see SINTEF's homepage at: http://www.sintef.no/ELCOM-90. From here you can download the latest version of all relevant documents as pdf-files for free.

This document contains ELCOM-90 Local Conventions for 2 new Functional Units made by SINTEF Energy Research and Powel ASA, 4 modifications made by Siemens and 8 modifications made by ABB. There is also a description of a Local Convention made in Argentine
The name of the 2 FU's are:

1) Initiator Data Transfer and 2) Retransmission of Historical Values

The "name" of the 4 modifications by Siemens are:

1)Command with Quality Flags. 2) Commanded Status Change Quality Flag
3) Double Precision Floating Point Value, 4)Fleeting Alarms

The "name" of the 8 modifications by ABB are:

Transmission of alarm states from the Responder
Unknown object
Supervisory Control blocked for Initiator
Supervisory Control blocked for Responder
Data collection blockade
Transducer out of range
Adaptation to FinELCOM standard
ELCOM-90 acceptance of ELCOM-83 Supervisory Control

The "Argentine" local convention is named: Millisecond representation
The FinELCOM Conventions version 1.3 is documented

SELECTED BY AUTHOR(S)		
	Data communication	Communication protocols
	Control centres	Energy management

TABLE OF CONTENTS

Page

0 VERSION HISTORY

Initial version plus one. Just some minor changes to responsible persons and references.

1. INTRODUCTION

The ELCOM-90 User Element Conventions [14], appendix B, permits Local Conventions to be established when communication requirements can not be fulfilled by the normal Functional Units (FU) and data types, specified in [14].

This report describes FU's developed by Powel ASA and the owners ABB, Siemens and SINTEF Energy Research during the period 1991 – 2003.

2. ASSOCIATED DOCUMENTS

2.1 ELCOM-83 documentation

[1]: TR 3522: **ELCOM-83 Application Service Definition**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1988-07-05

[2]: TR 3528: **ELCOM-83 Application Protocol Definition**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1988-07-14

[3]: TR 3523: **ELCOM-83 Definition of Local Application Interface**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1988-07-05

[4]: TR 3524: **ELCOM-83 Presentation Service Definition**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1988-07-06

[5]: TR 3527: **ELCOM-83 Presentation Protocol Definition**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1988-07-13

[6]: TR 3532: **ELCOM-83 Definition of Local Presentation Interface**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1988-09-12

[7]: TR 3649: **ELCOM-83 Conventions**
Norwegian Electric Power Research Institute, Trondheim, Norway, 1989-12-20
ISBN 82-594-0086-3

2.2 ELCOM-90 documentation

This document is one of a series of technical reports which form the complete ELCOM-90 documentation. Below you will find the numbers and titles for all the associated technical reports. New versions may be submitted when technical changes are made.
Please see SINTEF's homepage at: http://www.sintef.no/ELCOM-90. From here you can download the latest version of all relevant documents as pdf-files for free.

[8]: TR 3701: **ELCOM-90 Application Programming Interface Specification**

[9]: TR 3702: **ELCOM-90 Application Service Element. Service Definition**

[10]: TR 3703: **ELCOM-90 Application Service Element. Protocol Specification**

[11]: TR 3704: **ELCOM-90 Presentation Programming Interface Specification**

[12]: TR 3705: **ELCOM-90 Presentation Service Definition**

[13]: TR 3706: **ELCOM-90 Presentation Protocol Specification**

[14]: TR 3825: **ELCOM-90 User Element Conventions**

[15]: TR A3933: **ELCOM-90 Local Conventions**

[16] TR A4687: **PONG. The ELCOM net-watch procedure for TCP/IP networks**

[17] TR A4124: **ELCOM-90 Application Service Element, User's manual.**

[18] TR A6196: **Securing ELCOM-90 with TLS.**

3. DEFINITIONS AND ABBREVIATIONS

3.1 Definitions

ACCEPTOR address: The unique identification, octet string, of the responding service user.

Changing a group: Modifying one or more of the descriptor attribute values for an existing group identity.

Composite FU: Composite FUs act via invocation of other FUs by an Initiator UE only. They have no associated specific EASE service primitive sequence. Neither do they have any specific Responder part.

Configuration Set: The currently agreed-upon group configuration database shared between a number of INITIATOR/RESPONDER systems.

Configuring a group: Creating and defining a group.

Congestion error: Error situation in which the EASE is not able to receive *req.* or *res.* type service primitives, because of heavy traffic. General rules for handling congestion errors are given in chapter 4.2.1. Special rules per FU are given in the individual FU descriptions.

Coordinating Function: An Elcom User Element function that controls the local Functional Unit invocations.

Creating a group: Making a new group identity[1] legal, allocating a new group descriptor.

Defining a group: Attaching a set of implicitly numbered symbolic object identifiers to an empty group identity.

Deleting a group: Removing the group identity from the set of legal identities, deal locating the associated group descriptor.

Disrupting a Functional Unit or procedure:
Abruptly (non-orderly) terminating that Functional Unit or procedure.

Dynamic Association: An association between an INITIATOR UE and a RESPONDER UE, which may be created and terminated at the discretion of the INITIATOR UE.

EASE: Elcom Application Service Element.

Elcom partner: An Elcom site with which a given INITIATOR UE or RESPONDER UE may communicate via the EASE.

Elcom provider: The software component that implements the Elcom protocol in a given environment.

[1] Group number.

Elcom system: The set of User Elements, or single User Element, that utilise the Elcom provider that is addressed by the low-level part of a given Elcom address[2]. The rest of the local data processing environment of which the User Elements (or User Element), are (is) part, is also considered to belong to the same Elcom system.

Function Group: A named collection of Functional Units of related functionality.

Functional Unit invocation:
A specific instance of use of the Functional Unit.

Functional Unit type: A named collection of Functional Units of related action mechanisms.

Functional Unit, or Elcom User Element Functional Unit:
A named well-defined succession of EASE service primitives at the EAPIs of two communicating Elcom systems, constituting a single co-operative functional capability of an Elcom INITIATOR User Element and its peer Elcom RESPONDER User Element[3].

Group: A numbered set of named, and implicitly numbered, data objects in an Elcom system.

Incarnation: A consistent set of data values for a group or subgroup, all sampled at a given point in time.

INITIATOR address: The unique identification, octet string, of an Initiator User Element.

INITIATOR site: The collection of INITIATOR systems sharing a common Configuration Set. Equivalent to INITIATOR system, if no such sharing.

INITIATOR system: The collection of all INITIATOR UEs in a given Elcom system, together with the local data processing environment of which the collection is part.

INITIATOR User Element, or INITIATOR UE:
A User Element controlling associations, groups and data transfer, via the EASE.

Low-level Elcom address:
What is left of an Elcom address if the A-suffix character pair is removed.

Managing a group: Creating, changing or deleting a group.

Permanent Association:
An association between an INITIATOR UE and a RESPONDER UE, which is to be maintained at all times.

Primary FU: Primary FUs are the basic kind of FUs; these are characterized by their individual well defined sequence of EASE service primitives, and are always being invoked by an Initiator UE.

[2]An Elcom system may be addressed by more than one low-level Elcom address. For example, a number of different DTE numbers (which is one form of low-level Elcom addresses) will address the same Elcom system, if:
- All DTE numbers connect to X.25 lines attached to the device in which the Elcom system resides, and:
- All DTE numbers contain the single sub-address that is defined for Elcom-90.

Procedure: Sequence of prescribed actions in an Elcom INITIATOR UE and/or its peer RESPONDER UE.

Redefining a group: Modifying the existing set of object identifiers in a defined group.

RESPONDER system: The collection of all RESPONDER UEs in a given Elcom system, together with the local data processing environment of which the collection is part.

RESPONDER site: The collection of RESPONDER systems sharing a common Configuration Set. Equivalent to RESPONDER system, if no such sharing.

RESPONDER User Element:
The peer communications UE of an INITIATOR UE.

Secondary FU: Secondary FUs have individual well defined EASE service primitive sequences, but are always invoked by a Responder UE, as a result of local decision in that user element.

Subgroup: A contiguous range of objects within a group definition.

Transaction, or Elcom transaction:
A specific instance of use of an elementary EASE service.

User Element: The ELCOM User Element is defined as that part of the Elcom Application Entity that is not part of the EASE/EAPI. It may be of either the initiator type or of the responder type (see chapter 4.1).

3.2 Abbreviations

ADFU: Dynamic Association FU
AE: Application Entity
AP: Application Process
APFU: Permanent Association FU
ASE: Application Service Element
ATFU: Test Association FU
CS: Configuration Set
CS(I): The CS copy at the INITIATOR site
CS(R): The CS copy at the RESPONDER site
cnf.: confirm
DPFU: Periodic Data Transfer FU
DPRFU: Periodically Requested Data Transfer FU
DRFU: Requested Data Transfer FU
DSFU: Supervisory Control Data Transfer FU
DUFU: Unsolicited Data Transfer FU
DUMFU: Unsolicited Mixed Data Transfer FU
EAPI: Elcom Application Programming Interface
EASE: Elcom Application Service Element

[3]This is the basic definition. However, an FU may act exclusively through other FUs, having no specific EASE service primitive sequence associated with it. See about FU types, in chapter 4.2 "Functional Units (FUs)".

FU: (Elcom User Element) Functional Unit
GCFU: Group Configuration FU
GDFU: Group Definition FU
GMFU: Group Management FU
GRFU: Group Readout FU
ind.: *indication*
PDU: Protocol Data Unit.
RAFU: Restart Reactivate FU
req.: *request*
res.: *response*
RRFU: Restart Reconfigure FU
UE: User Element

4. FU DESCRIPTION TEMPLATE

NAME of FU

FUNCTION

CORDINATION rules
- Association usage

- Relation to other FUs
  - Invoking FUs
  - Invoked FUs
  - Disrupting FUs
  - Disrupted FUs

- Invocation
  - Prerequisites
  - Restrictions
  - Invoking events

- Termination
  - Orderly Termination
  - Disruption

- Procedures
  - EASE service primitives
  - Sequence
  - Parameter values

- Error handling
  - FU disruption
  - Illegal invocation attempt
  - Incoming EASE service primitive out of context
  - Timing errors
  - Congestion error
  - EASE service primitive parameter errors

5. CONTENTS IN APPENDIXES

The 2 Functional Unit (FU), the 4 modifications from Siemens, the 8 modifications from ABB and the Argentine change have got names and is described in a following Appendixes. The first two are fully described as FU's using the template listed in chapter 4. The modifications from Siemens and ABB are following another description template. The Argentine one is made as a special description made up from some correspondence between the Argentine company and SINTEF Energy Research.

The Appendix "number" and "name" of the 2 FU's are:

- A. Initiator Data Transfer
- B. Retransmission of Historical Values

The Appendix "number" and "name" of the 4 modifications made by Siemens are:

- C. Command with Quality Flags
- D. Commanded Status Change Quality Flag
- E. Double Precision Floating Point Value
- F. Fleeting Alarms

The Appendix "number" and "name" of the 8 modifications made by ABB are:

- G. Transmission of alarm states from the Responder
- H. Unknown object
- I. Supervisory Control blocked for Initiator
- J. Supervisory Control blocked for Responder
- K. Data collection blockade
- L. Transducer out of range
- M. Adaptation to FinELCOM standard
- N. ELCOM-90 acceptance of ELCOM-83 Supervisory Control

The Appendix "number" and "name" of the "Argentine" local convention is:

- O. Millisecond representation
- p. FinELCOM Conventions version 1.3

APPENDIX A - INITIATOR DATA TRANSFER FU

This appendix contains ELCOM-90 Local Conventions for a new Functional Unit (FU) used to transfer data telegrams from the Initiator to the Responder. The Initiator will establish the association as normal, but instead of asking for data, it will send data when activated.

This function is useful when data are available at random points of time, and when cost aspects makes it impossible to use the Unsolicited Data Transfer FU with permanent associations.

The name of the FU is «Initiator Data Transfer». The short name for the FU is DINFU.

Type: Primary.

A.1 FUNCTION

This FU transfers data in the opposite direction of the other data transfer FU's.

1. The Initiator sends data to the Responder, without any prior request for data, subject to the following general rules:
   - The data are associated with one specified ELCOM group only.
   - The sequence of the data values is the same as in the group definition.
   - The sequence of the data values may be a subset of the complete sequence, and the subset may vary from transmission to transmission.
   - For groups of type *Text-message-group*, each transmission shall contain the data value of exactly one object. In other words, for groups of this type each object shall constitute a data subset.
   - One single incarnation of the data value of any one object shall always be contained within a single transmission.
   - The Initiator UE must specify acknowledged or non-acknowledged operation, on a per transmission basis.

2. The Responder acknowledges reception of data, whenever such acknowledgement has been specified by the Initiator UE (More=False).

A.2 COORDINATION RULES

A.2.1 Association usage

All ELCOM interactions that are part of one invocation of the *Initiator Data Transfer FU* are conveyed by one single association. The association shall have the characteristics as specified in the section "Prerequisites", below.

A.2.2 Relation to other FUs

A.2.2.1 Invoking FUs

The Initiator Data Transfer FU shall not be invoked by another FU.

A.2.2.2 Invoked FUs

The Initiator Data Transfer FU shall not invoke any other FU.

A.2.2.3 Disrupting FUs

The Initiator Data Transfer FU may be disrupted by:

- *Permanent Association FU*
- *Dynamic Association FU*
- *Group Management FU*
- *Group Definition FU*
- *Restart Reconfigure FU*

A.2.2.4 Disrupted FUs

The *Initiator Data Transfer FU* shall not disrupt any other FU.

A.2.3 Invocation

A.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Initiator Data transfer FU*:

- *The Permanent Association FU or Dynamic Association FU*
- *The Group Configuration FU*

The *Permanent Association FU* or *Dynamic Association FU* invocation shall still be running at the time the *Initiator Data Transfer FU* is invoked, and in the case of the *Permanent Association FU*, the association maintained by the *Permanent Association* FU invocation shall be running (not temporarily broken) at the time.

The *Group Configuration FU* invocation shall be terminated when the *Initiator Data Transfer FU* is invoked.

The *Permanent Association FU* or *Dynamic Association FU* shall have been invoked in order to create (and for the *Permanent Association FU*, also to maintain) the association to be used for the interactions related to the current invocation of the *Initiator Data Transfer FU*.

A-suffices: Initiator UE: **CH**
Responder UE: **DH**

The *Group Configuration FU* shall have been invoked in order to define and configure the group that is to be transmitted, prior to the current invocation of the *Initiator Data Transfer FU*.

A.2.3.2 Restrictions

For any given INITIATOR/RESPONDER system combination, multiple simultaneous invocations of the *Initiator Data Transfer FU* for any given group are not allowed.

The *Initiator Data Transfer FU* must not be invoked while at least one of the following FUs are running, for the group involved:

- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*

A.2.2.3 Invoking events

The INITIATOR part of the *Initiator Data Transfer FU* may be invoked by:

- Local request via the Co-ordinating Function, the original source of which is outside the scope of this document.

Invocation of the RESPONDER part of the *Initiator Data Transfer FU* is attempted whenever a valid *A-Data-Transfer ind.* primitive is received via an association with the characteristics as defined for *Initiator Data Transfer FU*.

A.2.4 Termination

A.2.4.1 Orderly termination

Orderly termination of the INITIATOR part of a *Initiator Data Transfer FU* invocation may only be triggered by:

- Local request via the Co-ordinating Function, the original source of which is outside the scope of this document (premature termination).
- Local accumulated error count becoming too great. See "Error handling", below.
- Congestion error. See relevant section, below.
- Reception of an *A-Conf-Data ind.* service primitive after issuing an *A-Data (spont, More=False)* req. service primitive (normal termination).

The RESPONDER part of a *Initiator Data Transfer FU* invocation always terminates itself in an orderly manner (normal termination) upon reception of an A-Data (*Transmod=spont, More=False) ind.* primitive.

By termination, except for the case of congestion error, the RESPONDER UE shall try to issue an *A-Conf-Data req.* primitive.

A.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See section "Disrupting FUs", above.
- A fatal error condition encountered during operation. See section "Error handling", below.

A.3 PROCEDURES

A.3.1 EASE service primitives

The following elementary EASE services are used by the *Initiator Data Transfer FU*:

- *A-Data (spont)*
- *A-Conf-Data (spont)*

A.3.1.1 Sequence

The normal sequence of primitives is partitioned into 3 phases:

- **Phase 1: Data transmission**[4]

INITIATOR UE	EASE	RESPONDER UE
A-Data (spont,T) req.	-------- >	*A-Data (spont,T) ind.*
A-Data (spont,T) req.	-------- >	*A-Data (spont,T) ind.*
	.	
	.	
A-Data (spont,T) req.	-------- >	*A-Data (spont,T) ind.*
A-Data (spont,F) req.	-------- >	*A-Data (spont,F) ind.*

[1] Notation *T* and *F* signifies, for the parameter More-D, the values true and false, respectively.

In this phase, the following rules apply:

1) This document places no restriction on the number of consecutive *A-Data (spont,T) req*. primitives that may be issued before the terminating *A-Data (spont,F) req*. is issued. However, if too much time elapses between any *A-Data (spont,T)* primitive and the succeeding *A-Data (spont,T)* or *A-Data (spont,F)* primitive, an error situation occurs in the EASE. See the section "Error handling", below.

2) The INITIATOR UE is responsible for determining the timing of the individual *A-Data (spont) req*. primitives. However, the primitives shall be issued in without undue delay.

3) The rules stated in section "Function", above, also apply here.

Phase 2: Orderly terminating: Data acknowledgement.

INITIATOR	EASE	RESPONDER UE
A-Data (spont,T) req.	< --------	*A-Conf-Data (spont) req.*

The following rules apply:

1) The transition from phase 1 to phase 2 shall be triggered by the RESPONDER UE receiving and *A-Data (spont, F) ind*. primitive: The INITIATOR UE flags the fact that all requested data are transferred, by setting parameter More-D=false in the last transmission.

A.3.1.2 Parameter values

***A-Data (spont) req*. (INITIATOR)**

Parameter	req.
Gtype	Value *Measure-group*, *Status-group*, *Discrete-group*, *Logical-breaker status-group*, *Binary-command-group*, *Analog-setpoint-group*, *Digital-setpoint-group*, or *Text-message-group*, depending on the type of the group in question, as assumed in the CS(R).
Gnr	Reference number for the group in question, as assumed in the CS(R).
Transmod	= spontaneously
Index1[5]	Shall be > = 0: Equal to the lowest **Object number**, in accordance with the CS(R), of the range of **Object numbers** whose data values are contained in the Data parameter (below). Shall be equal to Index2, if Gtype = *Text-message-group*.[6]
Index2	Shall be > = value of Index1: Equal to the highest **Object number**, in accordance with the CS(R), of the range of **Object numbers** whose data values are contained in the Data parameter (below). Shall be equal to Index1, if Gtype = *Text-message-group*.

Table cont.

Parameter	req.
T	Time stamp, applying to the whole set of values contained in the Data parameter below, and determined by the INITIATOR UE. The use of UTC is recommended.
More-D	If current *A-Data (spont)* primitive is not the last carrying requested data, so that another will follow shortly: = *true*. If current *A-Data (spont)* primitive is the last carrying requested data: = *false.*
Data	The actual ELCOM data transferred. The structure is defined in Appendix A in ELCOM-90 User Element Conventions.
Result	= *result-ok*

[5] The number of values that shall be contained in the Data parameter can be computed as: Index2 – Index1 + 1.
[6] Only one object per transmission, for such groups

A-Conf-Data (spont) req. (RESPONDER)

Parameter	req.
Gtype	Copy of value from request (*A-Data of Phase* 1)
Gnr	Copy of value from request *(A-Data of Phase* 1)
Transmod	= *spontaneously*
Result	If no error detected by the RESPONDER UE: = *result-ok.* If error detected by the RESPONDER UE: Other value. See "Error handling" (below).

A.3.2 Error handling

A.3.2.1 FU disruption

Disruption by the Permanent Association *FU* or the *Dynamic Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Initiator Data Transfer FU* shall be triggered locally, as a part of the handling of incoming *A-P-Abort ind*. primitives in both the INITIATOR part and the RESPONDER part of the FU invocation handling the association on which the *Initiator Data Transfer FU* is running.

Both parts of the current invocation of the *Initiator Data Transfer FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself[7].

A.3.2.2 Illegal invocation attempt

FU not present:

If the *Initiator Data Transfer FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see section "Invocation", above. Consequently, the handling of this type of error is a local issue, outside the scope of this document.

If the *Initiator Data Transfer FU* is not present in a RESPONDER UE:

The RESPONDER User Element shall respond to activation attempts in one of two ways:

Either:
- Ignoring the incoming *A-Data ind*. primitive altogether

or:
- Issuing a *Conf-Data req*. primitive with Result = *remote-service-user-unavailable*.

[7] Local clean-up procedures are not specified by this document.

FU present, but attempt illegal:

Attempts of illegal invocations of the *Initiator Data Transfer FU* for any given group in an INITIATOR UE is a local issue, outside the scope of this document.

Attempts of illegal invocations of the *Requested Data Transfer FU* for any given group in a RESPONDER UE shall be handled by the RESPONDER UE in one of two ways:

Either:
- Ignoring the incoming *A-Data ind*. primitive altogether

or:
- Issuing an *A-Conf-Data req*. primitive with Result = *remote-service-user-unavailable*, without actually (re)-invoking the *Initiator Data Transfer FU* for the group concerned, in the RESPONDER UE.

A.3.2.3 Incoming EASE service primitive out of context

RESPONDER part:

State	***A-Data (spont.) ind.***
FU not running	Ignore, or: *Issue A-Conf-Data (spont.) req*., with Result = *remote-service-user-unavailable*.
FU running, waiting for A-Data (spont.) ind.	(Normal)

INITIATOR part:

State	***A-Conf-Data (spont.)ind.***
FU not running	Ignore, or local error indication/logging.
FU running, not waiting for *A-Conf-Data (spont) ind.*	Terminate FU invocation locally, then: local error indication/logging
FU running, waiting for *A-Conf-Data (spont.) ind.*	(Normal)

A.3.2.4 Timing errors

Error in RESPONDER part:

Error	Reaction from EASE	Specified action in FU
UE too late issuing *A-Conf- Data (spont.) req.* after receiving *A-Data (spont.,F) ind.*:	In RESPONDER: Local error from eventual attempt at *issuing A-Conf- Data (spont.) req.* In INITIATOR: *A-Conf-Data (spont.) ind.*, with Result = *remote- service-user-unavailable.*	RESPONDER part: Ignore, or local error indication/ logging, then proceed as normal. INITIATOR part: Terminate FU invocation locally, then: local error indication/logging

Error in INITIATOR part:

Error	Reaction from EASE	Specified action in FU
UE too late issuing next *A-Data (spont.) req.* after latest *A-Data (spont.,T) req.* issued	In INITIATOR: *A-Conf-Data (spont.) ind.*, with Result = *misbehaviour- of-local-service-user*. In RESPONDER: *A-Data (spont.) ind*., with Result = *remote-service-user unavailable.*	INITIATOR part: Terminate FU invocation locally. RESPONDER part: Terminate FU invocation locally.

A.3.2.5 Congestion error

RESPONDER part of the FU:
When occurring with an *A-Conf-Data (spont) reg*. attempt:
Terminate FU invocation locally.

INITIATOR part of the FU:
When occurring with an A-Data req. attempt:
Terminate FU invocation locally. (RESPONDER part of FU will eventually be terminated upon *A-Conf-Data (spont.) ind.* with Result ><result-ok, after time-out in the Elcom provider.)

A.3.2.6 EASE service primitive parameter errors

Errors in the A-Conf-Data (spont.) ind. primitive (detected by the INITIATOR part):

Error	Action in INITIATOR part of FU
Gtype value different from the value in CS(R) for the group associated with the current FU invocation. Result >< *result-ok* **For security class 2**: The received authentication code >< the generated authentication code based in the received data. **The security class 3**: The received checksum >< the generated checksum during the decipherment.	Error indication/logging, then terminate FU invocation, as normal. Error indication/logging, then terminate FU invocation, as normal *invalid-authentication-code-received* *decipherment-error*

Error in the ***A-Data (spont.) ind***. primitive (detected by the RESPONDER part):

The RESPONDER part shall in all cases, except for the case of syntax error in data parameter:

1. Terminate the current FU invocation in the INITIATOR by issuing an A-Conf-Data (spont.) req. with Result = <Value from table below>, Transmod = spontaneously, and the values of Gnr and Gtype as in the A-Data (spont.) ind. The data shall be ignored.

2. Terminate the RESPONDER part of the FU itself, optionally incrementing local error count and/or reporting or logging the error.

In the case of syntax error in Data parameter, "*invalid-authentication-code-received*" or "*decipherment-error*"[8], the RESPONDER part shall ignore the data, but otherwise proceed as normal, optionally incrementing local error count and/or reporting or logging the error.

Error req.	Value of parameter Result in *A-Conf-Data (spont.)*
Result = result-ok, and format error in T Result >< *result-ok*	*T-out-of-range* Same value as received parameter Result.

[8] See Appendix A in ELCOM-90 User Element Conventions for syntax definition.

APPENDIX B - RETRANSMISSION OF HISTORICAL VALUES

This Appendix contains Elcom-90 Local Conventions for a new Functional Unit (FU) used to perform Retransmission of Historical Values using an unsolicited data channel.

The Unsolicited Data Transfer FU (DUFU) is used to transfer online information only (Momentanous values), while Periodically Requested Data Transfer FU (DPRFU) is used to transfer values from the archives (Historical values). If values in the archive are updated after the DPRFU has been performed, the DUFU can not be used to retransmit those values, since values transmitted with DUFU are not stored in the archive. A new FU is therefore described here for this purpose.

The name of the FU is «Retransmission of Historical Values». The short name for the FU is DREFU.

Type: Primary

B.1 FUNCTION

This FU uses a permanent association to perform the data transfer.

A-suffices: Initiator Suffix: **CI**
Responder Suffix: **DI**

The FU is a copy of the Unsolicited Data Transfer FU, with the difference that no Initial Request shall be performed by the Initiator, and that received data shall be stored in the archive. Only floating point values may be transmitted with this FU (Object type=1).

- The Initiator establishes the association using the suffices described above.
- The Initiator sends Spontaneous Management (Start) for all groups that are containing objects for which retransmission of historical values shall take place.
- The Responder will transmit values when one of the objects in one the activated groups are updated. The actual updating of values, and the signalling between the Responder and the updating application, is not part of this document.
- The Responder will transmit one or several data telegrams to the Initiator. The last one will contain the parameter More=False.
- The Initiator will receive the data telegrams, storing the values in the archive.
- If the Initator receives a telegram with the parameter More=False, the Initiator will acknowledge the reception of data, sending a confirmation telegram.

The following rules also apply:

- The data are associated with one specified ELCOM group only.
- The sequence of the data values may be a subset of the complete sequence, and the subset may vary from transmission to transmission.
- One single incarnation of the data value of any one object shall always be contained within a single transmission.

- The Initiator UE must specify acknowledged or non-acknowledged operation, on a per transmission basis.

B.2 CORDINATION RULES

B.2.1 Association usage

All ELCOM interactions that are part of one invocation of the *Retransmission of Historical Values FU* are conveyed by one single association. The association shall have the characteristics as specified in the section "Prerequisites", below.

B.2.2 Relation to other FUs

B.2.2.1 Invoking FUs

The *Retransmission of Historical Values FU* shall not be invoked by another FU.

B.2.2.2 Invoked FUs

The *Retransmission of Historical Values FU* shall not invoke any other FU.

B.2.2.3 Disrupting FUs

The *Retransmission of Historical Values FU* may be disrupted by:

- *Permanent Association FU*
- *Group Management FU*
- *Group Definition FU*
- *Restart Reconfigure FU*

B.2.2.4 Disrupted FUs

The *Retransmission of Historical Values FU* shall not disrupt any other FU.

B.2.3 Invocation

B.2.3.1 Prerequisites

The following FUs shall have been invoked preceding any invocation of the *Retransmission of Historical Values FU:*

- The *Permanent Association FU*
- The *Group Configuration FU*

The *Permanent Association FU* invocation shall still be running at the time the *Retransmission of Historical Values FU* is invoked, and the association maintained by the *Permanent Association FU* invocation shall be running (not temporarily broken) at the time.

The *Permanent Association FU* shall have been invoked in order to create the association to be used for the interactions related to the current invocation of the *Retransmission of Historical Values FU*, with the following characteristics:

- A-suffix pair **CI** for the Initiator UE, and **DI** for the Responder UE.
- Spontaneous mode code as specified in the table in section «Spontaneous mode codes» in the Elcom-90 User Element Conventions.

The *Group Configuration FU* shall have been invoked in order to define and configure the group that is to be transmitted, prior to the current invocation of the *Retransmission of Historical Values FU*. The *Group Configuration FU* invocation shall be terminated when the *Retransmission of Historical Values FU* is invoked.

B.2.3.2 Restrictions

For any given INITIATOR/RESPONDER system combination, multiple simultaneous invocations of the *Retransmission of Historical Values FU* for any given group are not allowed.

The *Retransmission of Historical Values FU* must not be invoked while at least one of the following FUs are running, for the group involved:

- *Group Configuration FU*
- *Group Management FU*
- *Group Definition FU*

B.2.3.3 Invoking events

The INITIATOR part of the *Retransmission of Historical Values FU* may be invoked by:

- Local request via the Co-ordinating Function, the original source of which is outside the scope of this document.
- The Restart Reactivate FU.

Invocation of the RESPONDER part of the *Retransmission of Historical Values FU* is attempted whenever a valid *A-Spont-Mgnt (start) ind.* primitive is received via an association with the characteristics as defined for *Retransmission of Historical Values FU*.

B.2.4 Termination

B.2.4.1 Orderly Termination

Orderly termination of the INITIATOR part of a *Retransmission of Historical Values FU* invocation may only be triggered by:

- Local request via the Co-ordinating Function, the original source of which is outside the scope of this document (premature termination).
- Local accumulated error count becoming too great. See "Error handling", below.
- Congestion error. See relevant section, below.

Orderly termination of the Responder part of a *Retransmission of Historical Values FU* invocation may only be triggered by reception of a valid *A-Spont-Mgnt (stop) ind.* primitive.

B.2.4.2 Disruption

Both the INITIATOR and the RESPONDER part of a *Data Transfer FU* invocation may be disrupted by:

- Disruption of another FU invocation. See section "Disrupting FUs", above.
- A fatal error condition encountered during operation. See section "Error handling", below.

B.3.3 PROCEDURES

B.3.1 EASE service primitives

The following elementary EASE services are used by the *Retransmission of Historical Values FU*:

- *A-Spont-Mgnt*
- *A-Data (spont)*
- *A-Conf-Data (spont)*

B.3.1.1 Sequence

The normal sequence of primitives is partitioned into **3** phases:

Phase 1: Granting permission to send.

INITIATOR UE	EASE	RESPONDER UE
A-Spont-Mgnt (start) req.	---------->	*A-Spont-Mgnt (start) ind.*
A-Spont-Mgnt (start) cnf.	<----------	*A-Spont-Mgnt (start) res.*

Phase 2: Data transmission.[9]

INITIATOR UE	EASE	RESPONDER UE
A-Data (spont., T) ind.	<----------	*A-Data (spont., T) req.*
A-Data (spont., T) ind.	<----------	*A-Data (spont., T) req.*
	•	
	•	
A-Data (spont., T) ind.	<----------	*A-Data (spont., T) req.*
A-Data (spont., T) ind.	<----------	*A-Data (spont., T) req.*
A-Data (spont., T) ind.	<----------	*A-Data (spont., T) req.*
	•	
	•	
A-Data (spont., F) ind.	<----------	*A-Data (spont., F) req.*
A-Conf-Data (spont.) req.	---------->	*A-Conf-Data (spont.) ind.*
	•	
	•	

In this phase, the following rules apply:

1. For the current invocation of this FU, the *A-Conf-Data (spont.)* primitive is to be functionally interpreted as acknowledgement of the data carried by all preceding *A-Data (spont.)* primitives since the last *A-Conf-Data (spont.)* primitive.

2. This document places no restriction on the number of consecutive *A-Data (spont., T) req.* primitives that may be issued before an *A-Data (spont., F) req.* is issued. However, if too much time elapses between any *A-Data (spont., T)* primitive and the succeeding *A-Data (spont., T)* or *A-Data (spont., F)* primitive, an error situation occurs in the EASE. See the section "Error handling", below.

3. The RESPONDER UE is responsible for determining the timing and ordering of the *A-Data (spont.) req.* primitives for the different *Retransmission of Historical Values FU* invocations. The RESPONDER UE shall order its outgoing data queue according to the attribute **Priority Class** of the group[10] to which the data belongs:

- Data of a given **Priority Class** value may be sent prior to all pending data of greater **Priority Class** value.
- Data of equal **Priority Class** value may be sent in order of occurrence.
- **Priority Class** equal to zero shall be interpreted as "priority function off for these data", disabling priority check for the data concerned and always appending these at the end of the outgoing data queue.[11]

4. The RESPONDER UE may at any time choose to report data for any number of groups for which the *Retransmission of Historical Values FU* is invoked, via a local *Unsolicited Mixed Data Transfer FU* invocation instead of, or in addition to, the normal *A-Data (spont.) req.* primitives.

5. The rules stated in section "Function", above, also apply here.

[9]Notation: *T* and *F* signifies, for the parameter More-D, the values *true* and *false*, respectively.
[10]If data belong to more than one group, each occurrence of the data in the outgoing queue shall be handled independently, according the their individual **Priority Class** values.
[11]A RESPONDER UE altogether lacking support for the priority mechanism shall report the fact as an error whenever an INITIATOR UE tries to create or change a group into one for which **Priority Class** is different from zero.

Phase 3: Orderly termination: Withdrawing permission to send.

INITIATOR UE	EASE	RESPONDER UE
A-Spont-Mgnt (stop) req.	---------->	*A-Spont-Mgnt (stop) ind.*
A-Spont-Mgnt (stop) cnf.	<----------	*A-Spont-Mgnt (stop) res.*

The following rules apply:

1. The transition from phase 2 to phase 3 may follow either an *A-Data (spont., T)* primitive or an *A-Conf-Data (spont.)* primitive. If it follows an *A-Data (spont., T)* primitive, the error condition described in rule 2 of phase 2 above will eventually occur.

2. After an *Retransmission of Historical Values FU* invocation for a given group has been terminated, the RESPONDER UE must not try to invoke the *Unsolicited Mixed Data Transfer FU* for that group.

B.3.1.2 Parameter values

***A-Spont-Mgnt* :**

Parameter	***req.* (INITIATOR)**	***res.* (RESPONDER)**
Function	Phase 1: = *start* Phase 3: = *stop*	Copy of value from *ind.*
Gtype	Value *Measure-group*	Copy of value from *ind.*
Gnr	Reference number for the group in question, as defined in the CS.	Copy of value from *ind.*
Result	(Not applicable)	Action performed as specified. = *result-ok* Action not performed, due to error condition: Other value. See section "Error handling". (If Function = *start*, this means that the RESPONDER part of the *Unsolicited Data Transfer FU* has not been invoked, for the group in question).

A-Data (spont.,) req. (RESPONDER):

Parameter	***req.***
Gtype	Copy of value from *A-Spont-Mgnt (start) ind.* of Phase 1
Gnr	Copy of value from *A-Spont-Mgnt (start) ind.* of Phase 1
Transmod	= *spontaneous*
Index1[12]	Shall be > 0: Equal to the lowest **Object number**, in accordance with the CS(R), of the range of **Object numbers** whose data values are contained in the Data parameter (below).
Index2	Shall be >= value of Index1: Equal to the highest **Object number**, in accordance with the CS(R), of the range of **Object numbers** whose data values are contained in the Data parameter (below).
T	Time stamp, applying to the whole set of values contained in the Data parameter (below), and determined by the RESPONDER UE. The use of UTC is recommended.
More-D	If another *A-Data (spont.)* primitive for the group specified by the Gnr parameter (above) will follow shortly, so that data acknowledgement (*A-Conf-Data* primitive, issued by the INITIATOR) can be postponed: = *true*. If another *A-Data (spont.)* primitive for the group specified by the Gnr parameter (above) will NOT follow shortly, so that data acknowledgement (*A-Conf-Data* primitive) shall be issued by the INITIATOR: = *false*.
Data	The actual Elcom data transferred.
Length	Length of Data in objects. Shall be computed according to the length of the actual datatype, Index1 and Index2.
Result	= *result-ok*

A-Conf-Data (spont.) req. (INITIATOR):

Parameter	***req.***
Gtype	Copy of value from *A-Spont-Mgnt (start) ind.* of Phase 1
Gnr	Copy of value from *A-Spont-Mgnt (start) ind.* of Phase 1
Transmod	= *spontaneous*
Result	If no error detected by the INITIATOR UE: = *result-ok.* If error detected by the INITIATOR UE: Other value. See "Error handling" (below).

[12]The number of values that shall be contained in the Data parameter can be computed as: Index2 - Index1 + 1.

B.3.2 Error handling

B.3.2.1 FU disruption

Disruption by the *Permanent Association FU*:

Disruption of both the INITIATOR part and the RESPONDER part of the current invocation of the *Retransmission of Historical Values FU* shall be triggered locally, as a part of the handling of incoming *A-P-Abort ind.* primitives in both the INITIATOR part and the RESPONDER part of the *Permanent Association FU* invocation handling the association on which the *Retransmission of Historical Values FU* is running.

Both parts of the current invocation of the *Retransmission of Historical Values FU* shall be terminated gracefully, neither part attempting to issue any primitive associated with the termination itself.[13]

Following the termination, the *Permanent Association FU* will enter a state in which it will signal *Restart, spontaneous management lost* or *Restart, group management lost*, the next time an association with the characteristics for *Retransmission of Historical Values* is established between the same INITIATOR and RESPONDER UE pair.[14]

B.3.2.2 Illegal invocation attempt

FU not present:

If the *Retransmission of Historical Values FU* is not present in an INITIATOR UE:

Invocation requests are always generated locally; see section "Invocation", above. Consequently, the handling of this type of error is a local issue, outside the scope of this document.

If the *Retransmission of Historical Values FU* is not present in an RESPONDER UE:

The RESPONDER UE will not listen on the specified Suffix. The association will therefore never be established, and the FU will never be activated.

FU present, but attempt illegal:

Attempts of multiple simultaneous invocations of the *Retransmission of Historical Values FU* for any given group in an INITIATOR UE is a local issue, outside the scope of this document.

Attempts of multiple simultaneous invocations of the *Retransmission of Historical Values FU* for any given group in a RESPONDER UE shall be handled by the RESPONDER UE in one of two ways:
Either:

- Ignoring the incoming *A-Spont-Mgnt (start) ind.* primitive altogether

or:

[13]Local clean-up procedures are not specified by this document.
[14]The RESPONDER UE may be programmed to always signal the restart code "*Restart, spontaneous management lost*" during creation of associations for Unsolicited Data Transfer, regardless of previous history.

- Issuing an *A-Spont-Mgnt (start) res.* primitive with Result = *remote-service-user-unavailable* without actually (re-)invoking the *Retransmission of Historical Values FU* for the group concerned, in the RESPONDER UE.

B.3.2.3 Incoming EASE service primitive out of context

INITIATOR part:

State	***A-Spont-Mgnt (start) cnf.***	***A-Spont-Mgnt (stop) cnf.***	***A-Data (spont) ind.***
FU not running	Ignore, or local error indication/logging	Ignore, or local error indication/logging	Ignore, or: Simulate the entry of Phase 3 of an *Retransmission of Historical Values FU* invocation for the group in question, thus terminating the (supposed) current invocation of the *Retransmission of Historical Values FU* in the RESPONDER UE., or: Issue *A-Conf-Data (spont) req.*, with Result = *spontaneous-transfer-not-initiated*.
FU running, waiting for *A-Spont-Mgnt (start) cnf.*	(Normal)	(Parameter error)	Ignore, or local error indication/logging
Waiting for *A-Data (spont) ind.*	Ignore, or local error indication/logging	Ignore, or local error indication/logging	(Normal)
Waiting for *A-Spont-Mgnt (stop) cnf.*	(Parameter error)	(Normal)	Ignore, or local error indication/logging

RESPONDER part:

State	***A-Spont-Mgnt (start) ind.***	***A-Spont-Mgnt (stop) ind.***	***A-Conf-Data (spont) ind.***
FU not running	(Normal)	Ignore, or: Issue an *A-Spont-Mgnt (stop) res.* primitive with Result = *result-ok*, without terminating any *Retransmission of Historical Values FU* invocation in the RESPONDER UE.	Ignore, or local error indication/logging
FU running, not waiting for *A-Conf-Data (spont.) ind.*	Illegal invocation attempt; see relevant section.	(Normal) If Gnr is not active for spont. transfer, issue an *A-Spont-Mgnt (stop) res.* primitive with Result = *spontaneous-transfer-not-initiated*	(Normal)
FU running, waiting for *A-Conf-Data (spont.) ind.*	Illegal invocation attempt; see relevant section.	(Normal)	(Normal)

B.3.2.4 Timing errors

Error in INITIATOR part:

Error	**Reaction from EASE**	**Specified action in FU**
UE too late issuing *A-Conf-Data (spont.) req.* after receiving *A-Data (spont., F) ind.*:	In INITIATOR: Local error from eventual attempt at issuing *A-Conf-Data (spont) req.*	INITIATOR part: Ignore, or local error indication/logging, then proceed as normal.
	In RESPONDER: *A-Conf-Data (spont) ind.*, with Result = *remote-service-user-unavailable*.	RESPONDER part: Enter procedure for **Missing data acknowledgement** in RESPONDER. See section on parameter errors, below.

Error in RESPONDER part:

Error	Reaction from EASE	Specified action in FU
UE too late responding to *A-Spont-Mgnt (start) ind.*	In RESPONDER: Local error from eventual attempt at issuing *A-Spont-Mgnt (start) res.*	RESPONDER part: Terminate FU invocation locally.
	In INITIATOR: *A-Spont-Mgnt (start) cnf.*, with Result = *remote-service-user-unavailable*.	INITIATOR part: Terminate FU invocation locally.
UE too late issuing next *A-Data (spont.) req.* after latest *A-Data (spont., T) req.* issued	In RESPONDER: *A-Conf-Data (spont) ind.*, with Result = *misbehaviour-of-local-service-user*.	RESPONDER part: Enter procedure for **Missing data acknowledgement** in RESPONDER. See section on parameter errors, below.
	In INITIATOR: *A-Data (spont.) ind.*, with Result = *remote-service-user-unavailable*.	INITIATOR part: Local error indication/logging, then proceed as normal.
UE too late responding to *A-Spont-Mgnt (stop) ind.*	In RESPONDER: Local error from eventual attempt at issuing *A-Spont-Mgnt (stop) res.*	RESPONDER part: Terminate FU invocation locally.
	In INITIATOR: *A-Spont-Mgnt (stop) cnf.*, with Result = *remote-service-user-unavailable*.	INITIATOR part: Terminate FU invocation locally.

B.3.2.5 Congestion error

INITIATOR part of the FU:

When occurring with an *A-Spont-Mgnt (start) req.* attempt:
Terminate FU invocation locally.

When occurring with an *A-Conf-Data (spont) req.* attempt:
Enter Phase 3 (Orderly Termination) of the FU, attempting to issue *A-Spont-Mgnt (stop) req.*, with Result = *misbehaviour-of-local-service-user*.

When occurring with an *A-Spont-Mgnt (stop) req.* attempt:
Ignore, or notify operator

RESPONDER part of the FU:

When occurring with an *A-Spont-Mgnt (start) res.* attempt:

Terminate FU invocation locally. (INITIATOR part of FU will eventually be terminated after time-out in the Elcom provider.)

When occurring with an *A-Data req.* attempt:
Trigger local abrupt termination of supporting association[15]

When occurring with an *A-Spont-Mgnt (stop) res.* attempt:
Terminate FU invocation locally. (INITIATOR part of FU will eventually be terminated after time-out in the Elcom provider.)

B.3.2.6 EASE service primitive parameter errors

For all primitives except the *A-Spont-Mgnt (start) ind.* primitive, the parameter Gnr will always be valid within the context of this FU, *provided the FU is running*[16] for that group: It serves as identification of the FU invocation to which the incoming primitive shall be directed.

Errors in the ***A-Spont-Mgnt (start) ind.*** primitive (detected by the RESPONDER part):

Error	**Action in RESPONDER part of FU**
Gtype value not equal to value of attribute **Group type** in CS(R) for group no. Gnr. (The case of non-existing group no. Gnr is considered below.)	Issue *A-Spont-Mgnt (start) res.* with Result = *gtype-out-of-range*, and the values of Gnr and Gtype as in the corresponding *ind*. Do not invoke the FU.
Value of Gnr is illegal, or group no. Gnr does not exist in the CS(R).	Issue *A-Spont-Mgnt (start) res.* with Result = *gnr-out-of-range*, and the values of Gnr and Gtype as in the corresponding *ind*. Do not invoke the FU.
Value of Gtype is illegal.	Issue *A-Spont-Mgnt (start) res.* with Result = *gtype-out-of-range*, and the values of Gnr and Gtype as in the corresponding *ind*. Do not invoke the FU.
Group is created, but not defined	Issue *A-Spont-Mgnt (start) res.* with Result = *index-out-of-range*, and the values of Gnr and Gtype as in the corresponding *ind*. Do not invoke the FU.

[15]Such abrupt termination is effected by a local "detach" call (ADET) against the EAPI, followed by a local "attach" call (AATT).
[16]The case of the FU **not** being invoked for the group shall be handled outside the FU (by the Co-ordinating Function): The natural replying primitive shall be issued, with parameter Result = *gnr-out-of-range*.

Errors in the ***A-Conf-Data (spont) ind.*** primitive (detected by the RESPONDER part):

Error	**Action in RESPONDER part of FU**
Gtype value different from the value in CS(R) for the group associated with the current FU invocation	Enter procedure for **Missing data acknowledgement** (immediately below).
Result >< *result-ok*	Enter procedure for **Missing data acknowledgement** (immediately below).

Procedure for **Missing data acknowledgement**:

- If Gtype OK and Result = *spontaneous-transfer-not-initiated* (INITIATOR part not running):
  Terminate FU invocation locally
- If Gtype OK and Result = *misbehaviour-of-local-service-user* (RESPONDER data rate too slow, for
  More-D=T):
  Proceed as normal
- If Gtype not OK or Result any other value than *result-ok, spontaneous-transfer-not-initiated* and *misbehaviour-of-remote-service-user*:
  Retry, a locally determined number of times, including 0 (no retry) and infinite (always retry)
    - Wait, for a locally determined time span
    - Repeat all un-acknowledged *A-Data (spont) ind.*'s

  Terminate retry loop on any of the conditions (and actions) above, as well as Result = *result-ok*, in which case:
    Proceed as normal.

  If all retrials failed (Result ><*result-ok*, for all retrials), or local decision not to retry at all:
    Proceed as normal, or:
    Trigger abrupt termination of supporting association

Errors in the ***A-Spont-Mgnt (stop) ind.*** primitive (detected by the RESPONDER part):

Error	**Action in RESPONDER part of FU**
Gtype value different from the value in CS(R) for the group associated with the current FU invocation	Issue *A-Spont-Mgnt (stop) res.* with Result = *gtype-out-of-range*, and the values of Gnr and Gtype as in the corresponding *ind.* Do not terminate the FU.

Errors in the ***A-Spont-Mgnt (start) cnf.*** primitive (detected by the INITIATOR part):

Error	**Action in INITIATOR part of FU**
Mismatch between Gnr/Gtype/Function in primitive and Gnr/Gtype/Function in corresponding *A-Spont-Mgnt (start) req.*, and Result = *result-ok*.	Trigger Orderly Termination of the current FU invocation, issuing *A-Spont-Mgnt (stop) req.* for the group in question, with value of Gnr/Gtype/Function as in the corresponding *A-Spont-Mgnt (start) req.*
Result >< *result-ok*	Terminate the FU invocation locally

Errors in the ***A-Data (spont.) ind.*** primitive (detected by the INITIATOR part):

Error	**Action in INITIATOR part of FU**
Mismatch between Gtype in primitive and Gtype in corresponding *A-Spont-Mgnt (start) req.*.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
Invalid Index1 or Index2 (Validity conditions are given above, in section "Parameter values".)	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
Invalid Data (See Appendix A in this document for validity conditions)	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
Result = result-ok, and mismatch between Length and Index1/index2	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
Result = *result-ok*, and format error in T	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
Result >< *result-ok*	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count
The security class 2: The received authentication code >< the generated authentication code based in the received data.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.
For security class 3: The received checksum >< the generated checksum during the decipherment.	Ignore the data, but otherwise proceed as normal, optionally incrementing local error count.

Errors in the ***A-Spont-Mgnt (stop) cnf.*** primitive (detected by the INITIATOR part):

Error	**Action in INITIATOR part of FU**
Mismatch between Gtype in primitive and Gtype in corresponding *A-Spont-Mgnt (stop) req.*, and Result = *result-ok*.	Trigger abrupt termination of supporting association
Result >< *result-ok*	Trigger abrupt termination of supporting association

APPENDIX C – COMMAND WITH QUALITY FLAGS

C.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new feature which allows the RESPONDERS Quality Flags to be controlled by the INITIATOR system. Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

C.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

C.3 User Data Type

C.3.1 Binary command values

The Binary command has been modified so that the quality byte is used to transfer quality information along with the command. This extension allows a system to control the quality flags on the system which is providing the data. For example, an operator could send a command from the INITIATOR system to set the “Manually Entered” flag on the system providing the data (the RESPONDER) such that the “Manually Entered” quality flag is set on the provider. The change of quality flags would then trigger a spontaneous event which would set the “Manually Entered” quality flag on the system requesting the data (the INITIATOR). The quality flags are the same as listed for the Status Value.

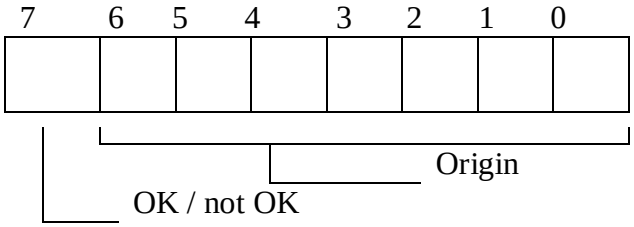
C.4 Quality Codes

Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data. For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

C.4.1 Status value quality codes

The quality byte for the Status value contains the following origin codes:

x0 000 0xx	Measured
x0 000 1xx	Manually entered
x0 001 0xx	Estimated
x0 001 1xx	Computed
x0 010 0xx	Held

The meanings of these terms are indicated as follows:

Measured — A point is "measured" when its value is acquired by one of several possible measuring methods (e.g. scanning).

Manually entered — The value of a point is "manually entered" when its current value was provided by input from an operator or dispatcher.

Estimated — A point is "estimated" when its value is calculated by a state estimator program.

Computed — A point is "computed" when its value is the result of a calculation using other data (scanned, computed, and/or estimated) as input variables.

Held — A numerical point whose value is measured is "held" when the most recent update was unsuccessful and an old value is held in the data base.

OK/Not OK — A point whose value is measured is "OK" when its value was acquired by the previous update; i.e., the point is not off-scan, and communications with the substation are successful.

A manually entered value is always "OK".

A point whose value is estimated by a state estimator is "OK" when the state estimator is running at its normally assigned frequency.

A point whose value is computed is "OK" when all the independent data points from which it is computed (measured, manually entered and/or estimated values) are "OK".

A point whose value is held is always "Not OK".

Bit 0 - 1 denotes the status value.

C.4.2 Binary command value quality codes

When the Binary command is issued the quality byte in the Command Indication is the same as listed for the Status Value. The quality flags sent with the Command Indication will be processed and set along with status. This is described in section C.3.

Once the command has been processed by the RESPONDER the quality byte is used to return information about the success of the command. The Command Confirm message uses the following origin codes:

x0 000 000	(OK)
x0 000 001	Object blocked at RTU side
x0 000 010	No connection to local device
x0 000 100	Command has illegal value
x0 000 101	Not authorized for supervisory control.

When the command has been handed over to the RESPONDER system's SCADA/EMS controlling function without failure, the value will be marked "OK". When the command could not be handed over to the RESPONDER system's SCADA/EMS controlling function or was rejected, the value is "Not OK" and is marked with one of the origin codes.

C.5 A-Data

Data : User data are of five types. See C.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See C.4).

Structure for Status-Group:

Quality Code and Status Value 1
. . .
Quality Code and Status Value n

C.6 A-Command-Transfer request

Data : User data are of three types. See C.3 for extensions to the standards:

Binary command group
Analogue setpoint group
Discrete setpoint group

Quality Code is set to match the quality of the point on the INITIATOR.

Structure for Binary Command-Group, Analogue Setpoint-Group and Discrete Setpoint Group:

Quality Code 1
Value 1
.
.
.
Quality Code n
Value n

APPENDIX D - COMMANDED STATUS CHANGE QUALITY FLAG

D.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new Commanded Status Change Quality Flag. This extension was found to be necessary so that dispatchers could be aware that a change was a commanded change and not a spontaneous event. Only areas which needed to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

D.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

D.3 User Data Types

D.3.1 Status values

The data type does not change for the implementation of this new quality flag. See section 9.4 for a description of the changes made to the quality flags.

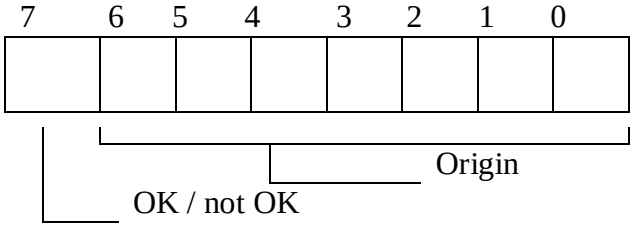
D.4 Quality Codes

Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data. For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

D.4.1 Status value quality codes

The quality byte for the Status value has been expanded to include a code signalling that the Status value changed as the result of a command issued by the operator. The present defined origin codes are:

x0 000 0xx	Measured
x0 000 1xx	Manually entered
x0 001 0xx	Estimated
x0 001 1xx	Computed
x0 010 0xx	Held
x0 011 0xx	Commanded

The meanings of these terms are indicated as follows:

Measured — A point is "measured" when its value is acquired by one of several possible measuring methods (e.g. scanning).

Manually entered — The value of a point is "manually entered" when its current value was provided by input from an operator or dispatcher.

Estimated — A point is "estimated" when its value is calculated by a state estimator program.

Computed — A point is "computed" when its value is the result of a calculation using other data (scanned, computed, and/or estimated) as input variables.

Held — A numerical point whose value is **measured** is "held" when the most recent update was unsuccessful and an old value is held in the data base.

Commanded — A point is "commanded" when its value has changed as the result of a command issued by the operator.

OK/Not OK — A point whose value is measured is "OK" when its value was acquired by the previous update; i.e., the point is not off-scan, and communications with the substation are successful.

A **manually entered** value is always "OK".

A point whose value is **estimated** by a state estimator is "OK" when the state estimator is running at its normally assigned frequency.

A point whose value is **computed** is "OK" when all the independent data points from which it is computed (measured, manually entered and/or estimated values) are "OK".

A point whose value is **held** is always "Not OK".

Bit 0 - 1 denotes the status value.

D.5 A-Data

Data : User data are of six types. See D.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values
Counter-Value Group	:	Double Precision Floating Point Value

Each value must be considered together with a quality code denoting its validity (See D.4).

Structure for Status-Group:

Quality Code and Status Value 1
. . .
Quality Code and Status Value n

APPENDIX E - DOUBLE PRECISION FLOATING POINT VALUE

E.1 Summary

This appendix describes the modification to the standard Elcom-90 specification SINTEF TR A3825 to add the new Double Precision Floating Point Value data type. This extension was found to be necessary so that higher precision values could be sent on the link. Only areas which needed to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

E.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

E.3 User Data Types

E.3.1 Double Precision Floating Point Value

A new User Defined Data Type was implemented in order to transfer values with higher precision than is available with the standard Elcom 90 floating point representation. The new data type is implemented using the IEEE Std 754-1985 64-bit double precision floating point format. This new User Data type is used with the new group type Double Precision Group Type (group type number = 100).

The data format is as follows:

Fraction:	52 bits. (0 <= f < 2)
Sign of the number:	1 bit.
Biased exponent:	11 bits. (-1021 <= e <= 1023)

One real value occupies 8 bytes in a value field:

Byte 0		Byte 1		...	Byte 7
1	11		52		
Sign	Exponent		Fraction		
msb		lsb	msb		lsb

For more information on the IEEE format see ANSI/IEEE 754.

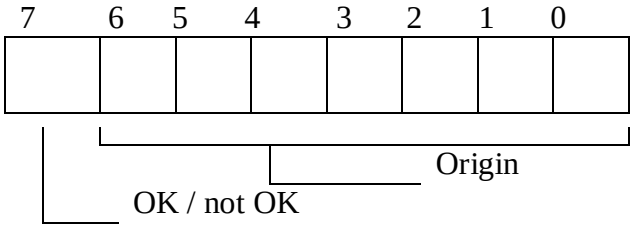
E.4 Quality Codes

Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data. For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

E.4.1 Double Precision Floating Point Value

The quality byte used by the Double Precision Floating Point Value data type is the same as the quality byte used by the standard Elcom-90 floating point value data type. The present defined origin codes are:

x0 000 000	Measured
x0 000 100	Manually entered
x0 001 000	Estimated
x0 001 100	Computed
x0 010 000	Held

The meanings of these terms are indicated as follows:

Measured — A point is "measured" when its value is acquired by one of several possible measuring methods (e.g. scanning).

Manually entered — The value of a point is "manually entered" when its current value was provided by input from an operator or dispatcher.

Estimated — A point is "estimated" when its value is calculated by a state estimator program.

Computed — A point is "computed" when its value is the result of a calculation using other data (scanned, computed, and/or estimated) as input variables.

Held — A numerical point whose value is **measured** is "held" when the most recent update was unsuccessful and an old value is held in the data base.

OK/Not OK — A point whose value is measured is "OK" when its value was acquired by the previous update; i.e., the point is not off-scan, and communications with the substation are successful.

A **manually entered** value is always "OK".

A point whose value is **estimated** by a state estimator is "OK" when the state estimator is running at its normally assigned frequency.

A point whose value is **computed** is "OK" when all the independent data points from which it is computed (measured, manually entered and/or estimated values) are "OK".

A point whose value is **held** is always "Not OK".

E.5 A-Data

Data : User data are of six types. See E.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values
Double Precision Group	:	Double Precision Floating Point Value

Each value must be considered together with a quality code denoting its validity (See E.4).

Structure for Real (Measure)-Group, Logical Breaker Status-Group, Discrete-Group, and Double Precision Floating Point Values:

Quality Code 1
Value 1
.
.
.
Quality Code n
Value n

APPENDIX F - FLEETING ALARMS

F.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the special handling for Fleeting Alarms. This extension was found to be necessary since fleeting alarms do not have a state. Fleeting alarms are event which only occur spontaneously. Only areas which needed to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

F.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

F.3 User Data Types

F.3.1 Status values

If a fleeting alarm indicator is included in a group being requested by a remote partner then during the general interrogation the value will be sent as “Off”. The only time that a value of “On” is transferred for these fleeting alarms is with a spontaneous change.

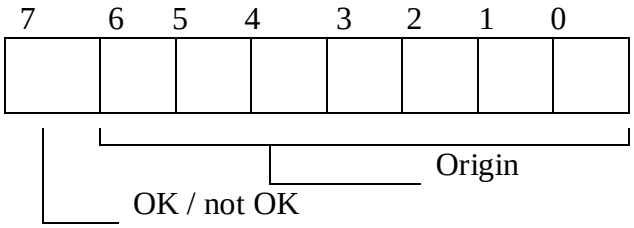
F.4 Quality Codes

Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data. For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

F.4.1 Status value quality codes

The quality byte for the Status value has been expanded to include a code signalling that the Status value changed as the result of a command issued by the operator. The present defined origin codes are:

x0 000 0xx	Measured
x0 000 1xx	Manually entered
x0 001 0xx	Estimated
x0 001 1xx	Computed
x0 010 0xx	Held
x0 011 0xx	Commanded

The meanings of these terms are indicated as follows:

Measured — A point is "measured" when its value is acquired by one of several possible measuring methods (e.g. scanning).

Manually entered — The value of a point is "manually entered" when its current value was provided by input from an operator or dispatcher.

Estimated — A point is "estimated" when its value is calculated by a state estimator program.

Computed — A point is "computed" when its value is the result of a calculation using other data (scanned, computed, and/or estimated) as input variables.

Held — A numerical point whose value is **measured** is "held" when the most recent update was unsuccessful and an old value is held in the data base.

Commanded — A point is "commanded" when its value has changed as the result of a command issued by the operator.

OK/Not OK — A point whose value is measured is "OK" when its value was acquired by the previous update; i.e., the point is not off-scan, and communications with the substation are successful.

A **manually entered** value is always "OK".

A point whose value is **estimated** by a state estimator is "OK" when the state estimator is running at its normally assigned frequency.

A point whose value is **computed** is "OK" when all the independent data points from which it is computed (measured, manually entered and/or estimated values) are "OK".

A point whose value is **held** is always "Not OK".

F.5 A-Data

Data : User data are of five types. See F.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See 11.4).

Structure for Status-Group:

Quality Code and Status Value 1
. . .
Quality Code and Status Value n

APPENDIX G – TRANSMISSION OF ALARM STATES FROM THE RESPONDER

G.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new Transmission of alarm states from the Responder. This extension was found to be necessary to transmit alarm level for a measurement, from the Responder to the Initiator. This is achieved by implementing new quality flags.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

G.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

G.3 User Data Types

G.3.1 Real values

The data type does not change for the implementation of this new quality flag. See section G.4 for a description of the changes made to the quality flags.

G.4 Quality codes

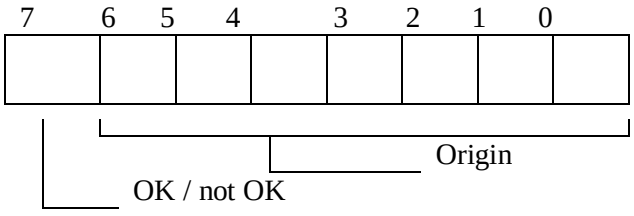
Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data.

For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

G.4.1 Real value quality codes

The quality byte for the Real value has been expanded to include a code signalling that the value is to be regarded as an alarm. The present defined origin codes are:

x0 000 0xx	Measured
x0 000 1xx	Manually entered
x0 001 0xx	Estimated
x0 001 1xx	Computed
x0 010 0xx	Held
00 100 000	Alarm implemented, Normal value
00 110 000	Alarm implemented, Low alarm
00 110 001	Alarm implemented, Low warning
00 110 010	Alarm implemented, High warning
00 110 011	Alarm implemented, High alarm

The meanings of these terms are indicated as follows:

Measured — A point is "measured" when its value is acquired by one of several possible measuring methods (e.g. scanning).

Manually entered — The value of a point is "manually entered" when its current value was provided by input from an operator or dispatcher.

Estimated — A point is "estimated" when its value is calculated by a state estimator program.

Computed — A point is "computed" when its value is the result of a calculation using other data (scanned, computed, and/or estimated) as input variables.

Held — A numerical point whose value is **measured** is "held" when the most recent update was unsuccessful and an old value is held in the data base. Attention: When Held is used, the value is always regarded as NOT OK (Bit 7).

Alarm implemented — When bit 5 is one, and bit 7 is 0, this value is reported with an alarm state. (This can not be in conflict with “Held”, because bit 7 is always 1 when “Held” is set). Bit 4=1 means that the object is in an alarm state. Bit 4=0 means that the object is not in an alarm state. The alarm state itself is set in bit 0 and 1.

OK/Not OK — A point whose value is measured is "OK" when its value was acquired by the previous update; i.e., the point is not off-scan, and communications with the substation are successful.

A **manually entered** value is always "OK".

A point whose value is **estimated** by a state estimator is "OK" when the state estimator is running at its normally assigned frequency.

A point whose value is **computed** is "OK" when all the independent data points from which it is computed (measured, manually entered and/or estimated values) are "OK".

A point whose value is **held** is always "Not OK".

G.5 A-Data

Data : User data are of five types. See G.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See G.4).

Structure for Status-Group:

Quality Code and Status Value 1
.
.
.
Quality Code and Status Value n

APPENDIX H: LOCAL CONVENTIONS – UNKNOWN OBJECT

H.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new quality code "Unknown object" signalled from the Responder. This extension was found to be necessary to signal that the object name is not valid when the data transfer is active. This is achieved by implementing new quality flags.

Some responders are implemented such that they always accept every object names in the Group Configuration FU. The check on legal object name is therefore postponed until the Data Transfer FU when a request for values is sent to the RTU. It is therefore necessary to signal that the reason for missing values is due to an unknown object in the RTU.

Only areas which needed to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

H.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

H.3 User Data Types

This new quality code applies to all standard data types.

H.4 Quality codes

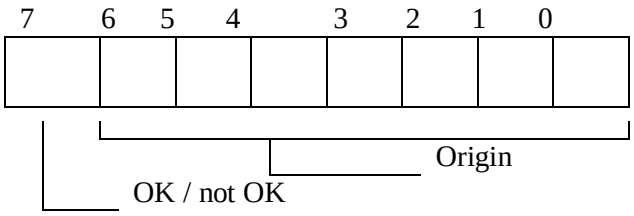
Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data.

For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

H.4.1 Additional quality code for "Unknown Object"

The quality byte for all data types has been expanded to include a code signaling that the value is not given because the requested object was unknown in the RTU. The new origin code is:

x1 xxx xxx	Unknown Object

The meanings of these terms are indicated as follows:

Unknown Object Some responders are implemented such that they always accept every object names in the Group Configuration FU. The check on legal object name is therefore postponed until the Data Transfer FU when a request for values is sent to the RTU. It is therefore necessary to signal that the reason for missing values is due to an unknown object in the RTU.

OK/Not OK An **Unknown Object** is always "Not OK".

H.5 A-Data

Data : User data are of five types. See H.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See H.4).

Structure for Status-Group:

Quality Code and Status Value 1
.
.
.
Quality Code and Status Value n

APPENDIX I - SUPERVISORY CONTROL BLOCKED FOR INITIATOR

I.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new quality code “Supervisory Control Blocked for Initiator”. This extension was found to be necessary in an implementation for Sydkraft, Banverket and CELESC. This is achieved by implementing new quality flags.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

I.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

I.3 User Data Types

The new quality code applies to user data type 2, Status values.

I.4 Quality Codes

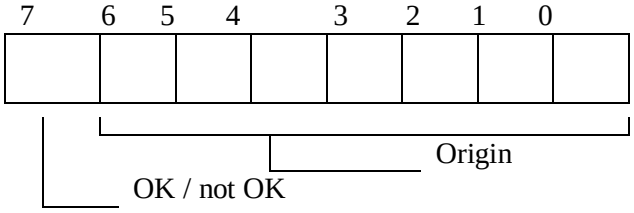
Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data.

For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

I.4.1 Additional quality code for "Supervisory Control blocked for Initiator"

xx x1x xxx Supervisory Control blocked for Initiator

The meanings of these terms are indicated as follows:

Supervisory Control blocked for Initiator This object can not be controlled by the Initiator.

I.5 A-Data

Data : User data are of five types. See I.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See I.4).

Structure for Status-Group:

Quality Code and Status Value 1
.
.
.
Quality Code and Status Value n

APPENDIX J - SUPERVISORY CONTROL BLOCKED IN RESPONDER

J.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new quality code “Supervisory Control Blocked in Responder”. This extension was found to be necessary in an implementation for Sydkraft, Banverket and CELESC. This is achieved by implementing new quality flags.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

J.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

J.3 User Data Types

The new quality code applies to user data type 1 (Real value), and user data type 2 (Status values).

J.4 Quality Codes

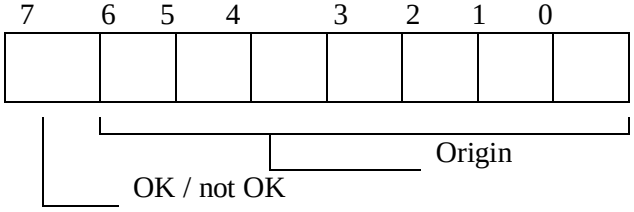
Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data.

For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

J.4.1 Additional quality code for "Supervisory Control blocked in Responder"

xx 1xx xxx Supervisory Control blocked in Responder

The meanings of these terms are indicated as follows:

Supervisory Control blocked in Responder This object can not be controlled by the Responder.

J.5 A-Data

Data : User data are of five types. See J.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See J.4).

Structure for Real (Measure)-Group, Logical Breaker Status-Group, Discrete-Group:

Quality Code 1
Value 1
.
.
.
Quality Code n
Value n

Structure for Status-Group:

Quality Code and Status Value 1
.
.
.
Quality Code and Status Value n

APPENDIX K - DATA COLLECTION BLOCKED

K.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new quality code “Data Collection Blocked”. This extension was found to be necessary in an implementation for Banverket. This is achieved by implementing one new quality flag.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

K.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

All arrays are octet arrays.

K.3 User Data Types

The new quality code applies to user data type 1 (Real value), and user data type 2 (Status values).

K.4 Quality Codes

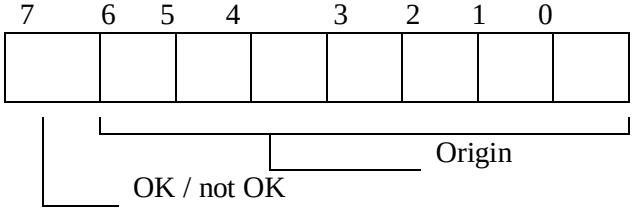
Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data.

For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

K.4.1 Additional quality code for "Data Collection Blocked"

x1 xxx xxx	Data Collection Blocked

The meanings of these terms are indicated as follows:

Data Collection Blocked	Data collection is blocked for this object.

K.5 A-Data

Data : User data are of five types. See K.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See K.4).

Structure for Real (Measure)-Group, Logical Breaker Status-Group, Discrete-Group:

Quality Code 1
Value 1
.
.
.
Quality Code n
Value n

Structure for Status-Group:

Quality Code and Status Value 1
.
.
.
Quality Code and Status Value n

APPENDIX L - TRANSDUCER OUT OF RANGE

L.1 Summary

This appendix describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the new quality code “Transducer out of range”. This extension was found to be necessary in an implementation for REMU. This is achieved by implementing one new quality flag.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

L.2 Structure

The octets in this appendix are numbered starting from 0 and increasing in order of transmission. The bits in an octet are numbered from 0 to 7, where bit 0 is the low-ordered bit.

All octets are numbered in decimal. All values are given in decimal when nothing else is stated. Codes are given in binary. All parameters are represented in twos complement integer when nothing else is stated.

Integer values represented in two octets have their least significant part stored in the octet with the highest octet no.

L.3 User Data Types

The new quality code applies to user data type 1 (Real value).

L.4 Quality Codes

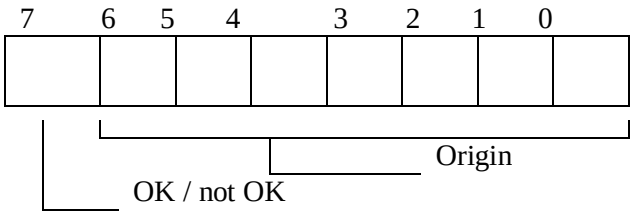
Each value transmitted, except the text message strings, is delivered together with a quality code denoting the quality and origin of the value.

For all values, except the status values, the quality code is delivered in a separate octet. For status values the quality code is coded in the most significant bits of the octet.

The most significant bit of the octet is used to express the validity of the corresponding value. If it is 0 the value is regarded OK, else it is regarded not OK.

For status values bit 2 - 6 are used to express the origin of the data.

For other values bit 0 - 6 are used to express the origin of the data.



OK code: 0 - OK
1 - Not OK

L.4.1 Additional quality code for "Transducer out of range"

x1 xxx xxx	Transducer out of range

The meanings of these terms are indicated as follows:

Transducer out of range: Transducer is out of range.

L.5 A-Data

Data : User data are of five types. See L.3 for extensions to the standards:

Real (Measure)-Group	:	Floating Point Values
Discrete-Group	:	Integer Values
Status-Group	:	Binary Values
Logical Breaker Status Group	:	Binary Values
Text Message Group	:	ASCII Values

Each value must be considered together with a quality code denoting its validity (See L.4).

Structure for Real (Measure)-Group, Logical Breaker Status-Group, Discrete-Group:

Quality Code 1
Value 1
.
.
.
Quality Code n
Value n

APPENDIX M - ADAPTATION TO FINELCOM STANDARD

M.1 Summary

This appendix shortly describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to implement the adaptation to the FinElcom standard. The FinElcom standard is used in Finland against Elcom-83 partners.

For the full description of the FinElcom standard, see appendix P. Appendix P is generated using a scanner on the original paper document from FinELCOM SOFTWARE. It may therefore contain some errors.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

M.2 Password

Part of the User Data field is used to represent password during connection establishment.

M.3 Result Codes

Result codes different from the standard Elcom Result codes are used.

M.4 Suffices

Suffices different from the standard Elcom suffices are used by the Responder.

M.5 Restart code

The Restart Code (part of the User Data field in A-Connect Response) is given as number instead of ASCII.

APPENDIX N - ELCOM-90 ACCEPTANCE OF ELCOM-83 SUPERVISORY CONTROL

N.1 Summary

This appendix shortly describes the modifications to the standard Elcom-90 specification SINTEF TR A3825 needed to accept Supervisory Control from an Elcom-83 Class 2, Version 0 partner.

Only areas which need to be described were included in this specification. All areas left unaffected are left to the SINTEF specifications. See these specifications for a more detailed reference.

N.2 Modifications

Some customers have implemented Supervisory Control Functional Unit (FU) in Elcom-83. According to the documentation, this is not legal. Some modifications in the Elcom API, and the Initiator and the Responder, have to be done to allow this combination of Class/Version and FU.

The modifications in the code consists of removing restrictions in the error handling part of the code, so that Supervisory Control functions against Elcom-83 partners will be allowed.

APPENDIX O - LOCAL CONVENTIONS MADE IN ARGENTINE

O.1 Encoding of milliseconds.

In the ELCOM-90 specifications version .01 there are some places references to *twos complement* representation of integers. Due to some formulations that may be a little ambiguous, there have been some different interpretations of the specifications regarding the encoding of milliseconds, and thus there are implementations of the system having some unintended features. To avoid further problems the actual formulations are rewritten in version .02 of the specifications.

The reference code of ELCOM-90 is reviewed to be sure that the reference code is not affected by this misunderstanding. The reference code comply with version .02 of the ELCOM specifications.
CAMMESA and four independent companies from Argentina have developed ELCOM implementations, encoding the milliseconds in the following way:

"The milliseconds number is controlled against the range 0-999, complemented (65536-this number) and this result is put into the PDU. "

This solution does not comply with the specifications version .02.

In order to solve part of the problem when communicating with implementations that comply with the reference version, the software from the companies mentioned above, has been modified to accept complemented and non-complemented numbers. The milliseconds in received PDUs are accepted either if they fits in the range 0-999, or in its twos complement. Of course the values are passed to the application levels encoded in a unique way. But in the transmitted PDUs, the milliseconds are always complemented.

APPENDIX P - FINELCOM CONVENTIONS VERSION 1.3

The FinELCOM Server Software follows mainly the conventions defined by ELCOM WG in the document Conventions for ELCOM-83 applications 28.7.1989. Some additions to these conventions has been defined to implement functions which are not supported by the ELCOM-83 protocol. Additions are listed below.

P.1 The implementation of the selective-cyclic transfer mode

Selective-cyclic is defined as cyclic but the second User-data octet gets the value 10.

Second User-data octet:

0	-	no cyclic
1	-	cyclic
10	-	selective-cyclic.

The definition and the handling of the data are like in Cyclic Spontaneous Transmission (Conventions for ELCOM-83 Applications, section 2.6). The only difference is that only the data which has changed it's value is transferred.

P.2 Suffixes

Responder use a general suffix '00' to receive all connect requests.

In addition to the suffixes define in conventions by ELCOM WG (Conventions for ELCOM-83 Applications, Section 2.0) the next suffixes are used:

Function	Initiator	Responder
Group configuration	CA	
Selective-cyclic data transfer	CC	DC
Special group inquiry	CP	DP
Manual inquiries	CQ	DR
Real time data	CR	DR

P.3 The additional reason/result codes.

128 command ok
129 command not ok
130 command not acknowledged
131 set value ok
132 set value not ok
133 set value not acknowledged

134 incompatible group type
135 privilege violation
136 incompatible time
137 remote locked
138 remote limit reached
139 group not exists
140 group not valid
141 remote cross reference failure
142 remote failure

P.4 Commands and set values

Commands and set Values are implemented by using the spontaneous transfer mode. The system, which will receive the commands/set values, defines the group which consists command/set value data items and defines this group into the spontaneous transfer mode. The data type of the group is a floating point type. The data items must also be known commands/set values in the responding system. In case of data type conflict, the result code incompatible group type is returned in the A_Data_Confirmation message.

The responder sends commands/set values as a spontaneous data and the acknowledgement status is returned by the initiator in the result-field of the confirmation. The next additional result/reason codes are used:

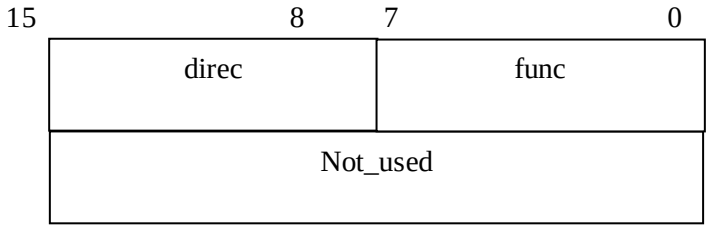
- command ok
- command not ok
- command not acknowledged.

Depending on the implementation the initiator can get these result codes from the local SCADA system or evaluate them by itself.

Only one active (not acknowledged) command/set value is allowed for one system at the same time.

It is on the Initiator's responsibility to create a logical channel for the spontaneous data transfer used for the commands and set values (see section 2.4 Spontaneous Transmission in conventions). The Responder's responsibilities are the same as in the case of the spontaneous transfer mode.

The floating point data has the following interpretation when used as a control data item:



direc : direction of the command

1 - off
2 – on

func : function of the command

1 - select
2 - execute
3 - cancel
252 - immediate execute

Set values are sent as the normal floating point data types.

P.5 Group configuration

Additional result codes are used with group management definition services:

- If the data item is not allowed to the initiating system the result code privilege violation is returned.
- If the local representation of the data items is unknown in the responder site the result code remote cross reference failure is returned.
- If the maximum number of groups allowed to the initiating system is exceeded, the responder returns the result code remote limit reached.

In all of the these cases the handled group is deleted in the responding system.

P.6 History data transfer

The following restrictions are in force with the historic data transfer (A_Init_Data_Transfer request, TO(1) <> -1):

- Only integer and real type data groups can be requested.
- Only one data object is allowed for one AInit-Data-Transfer request.

If the responding system can't find the requested history levels the A-Data result code incompatible time is returned.

P.7 Access control

The 8 last octets 6..13 of the User-data field in the connect request service are reserved for the password. The password must include exactly 8 octets. Before accepting the call the responder checks the password in the request against the password given for the system in the local data base.

There can be one different password for each remote systems both in the initiator and responder site. The initiator uses this password in all connect requests which are sent to that system. The responder

accepts the connect request if the password used by the initiator is the same as defined in the responder's site for the calling system.

If the password is not matching the responder returns the result code privilege violation in the connect response. The responding system can reject an incoming connect request by returning the additional result code remote locked.

P.8 Resource Control

The responder controls the resources allowed to each remote initiating system. If these resources are exceeded the responder returns an additional result code remote limit reached (A_Spont_Mgnt, A_Group_Mgnt).

P.9 Spontaneous transfer modes

Only one transfer mode based on spontaneous transfer (spontaneous, cyclic, selective-cyclic) is allowed for each group at the time.