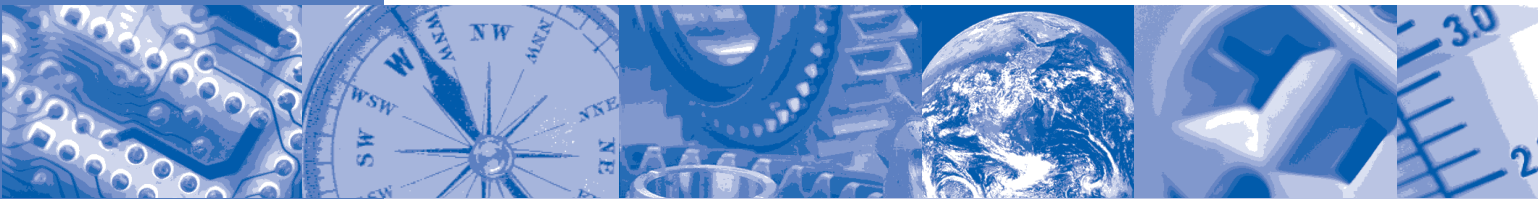


ANSI/ISA–50.02, Part 6–1998

Formerly ANSI/ISA–S50.02, Part 6–1998



Fieldbus Standard for Use in Industrial Control Systems Part 6: Application Layer Protocol Specifications



Approved 15 November 1998

ANSI/ISA-50.02, Part 6-1998, Fieldbus Standard for Use in Industrial Control Systems
Part 6: Application Layer Protocol Specification

ISBN: 1-55617-686-4

Copyright © 1998 by the Instrument Society of America (ISA). All rights reserved. Not for resale. No part of this publication may be reproduced in any form, including an electronic retrieval system, without the prior written permission of the copyright holders. All requests pertaining to the standard should be submitted to ISA.

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, North Carolina 27709

Preface

This preface, as well as all footnotes and annexes, is included for informational purposes and is not part of ANSI/ISA-50.02, Part 6-1998.

This Standard has been prepared as part of the service of the ISA, the international society for measurement and control, toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static, but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms, and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards, recommended practices, and technical reports. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards to the greatest extent possible. *Standard for Use of the International System of Units (SI): The Modern Metric System*, published by the American Society for Testing & Materials as IEEE/ASTM SI 10-97, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of the ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of the ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

CAUTION — ISA ADHERES TO THE POLICY OF THE AMERICAN NATIONAL STANDARDS INSTITUTE WITH REGARD TO PATENTS. IF ISA IS INFORMED OF AN EXISTING PATENT THAT IS REQUIRED FOR USE OF THE STANDARD, IT WILL REQUIRE THE OWNER OF THE PATENT TO EITHER GRANT A ROYALTY-FREE LICENSE FOR USE OF THE PATENT BY USERS COMPLYING WITH THE STANDARD OR A LICENSE ON REASONABLE TERMS AND CONDITIONS THAT ARE FREE FROM UNFAIR DISCRIMINATION.

EVEN IF ISA IS UNAWARE OF ANY PATENT COVERING THIS STANDARD, THE USER IS CAUTIONED THAT IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS THAT MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA IS NOT RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER'S INTENDED APPLICATION.

HOWEVER, ISA ASKS THAT ANYONE REVIEWING THIS STANDARD WHO IS AWARE OF ANY PATENTS THAT MAY IMPACT IMPLEMENTATION OF THE STANDARD NOTIFY THE ISA STANDARDS AND PRACTICES DEPARTMENT OF THE PATENT AND ITS OWNER.

ADDITIONALLY, THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER'S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

The following members of ISA Subcommittee SP50.5 contributed to the development of this document:

NAME	COMPANY
U. Doebrich, Chairman	Siemens AG
H. Dammeyer, Managing Director	The Ohio State University
G. Wood, Managing Director	Graeme Wood Consulting
C. Akiyama	Yokagawa Electric Corporation
W. Borst	Borst Automation
*P. Brett	Honeywell, Inc.
R. Crowder	Ship Star Associates Inc.
E. dela Hostria	Rockwell Automation
M. Giraudi	Elsag Bailey
*K. Krivoshein	Fisher Controls International Inc.
*Y. Kumeda	Yamatake Industrial Systems
T. Laine	Cegelec
K. Lindner	Endress + Hauser GmbH
O. Mirabella	Universita Degli Studi di Catania
*L. Neitzel	Fisher•Rosemount Systems Inc.
S. Pedersen	ABB Corporate Research
F. Russo	ENEL-Ricerca/PEA
K. Schwarz	Schwarz Communication Consult
E. Skabowski	Tendent Consulting
P. Thiessmeier	Klockner-Moeller
N. Tobol	Ronan Engineering
C. Williams	Eastman Kodak Company
G. Winchester	National Electrical Manufacturing Assoc.

* One Vote per Company

The following members of ISA Committee SP50 contributed to the development of this document:

NAME	COMPANY
R. Caro, Chairman	Automation Research Corporation
H. Dammeyer, Managing Director	The Ohio State University
G. Wood, Managing Director	Graeme Wood Consulting
C. Akiyama	Yokogawa Electric Corporation
J. Aliperti	Chromatech Consultoria Limited
W. Baker	KML, Inc.
L. Beckman	HIMA Americas, Inc.
M. Borges	Petrobras Cenpes Supen Diprom
*R. Bowden	Fisher-Rosemount Ltd.
S. Boyer	Suncor Energy Inc.
R. Breithaupt	Siemens
*P. Brett	Honeywell, Inc.
P. Burton	Peter Burton & Associates
R. Canfield	Square D Company
A. Capel	Comgate Engineering, Ltd.
B. Casner	Miller Electric Manufacturing Company
*S. Cavalieri	Universita Degli Studi di Catania
R. Crowder	Ship Star Associates, Inc.
E. dela Hostria	Rockwell Automation
*U. Doebrich	Siemens AG
J. Dohoney	Jet Propulsion Laboratory
B. Dumortier	Schneider Electric SA
*J. Elias	Honeywell, Inc.
T. Feindel	R. Stahl, Inc.
M. Giraudi	Elsag Bailey
J. Gray	Siebe Field Measurement and Control
*P. Griem, Jr.	Honeywell IAC
B. Gross	General Motors Corporation
*C. Gross	Dow Chemical Company
A. Gupta	Ficon Technology
H. Hasler	Adelcontrol
*T. Ito	Fuji Electric
*V. Jacobson	Ronan Engineering Company
J. Jamison	Bantrel Inc.
*K. Kakizakai	Fuji Electric Company, Ltd.
R. Ketcham	Union Carbide Corporation
*K. Krivoshein	Fisher Controls International Inc.
Y. Kumeda	Yamatake Industrial Systems
T. Kuusisto	Consultant
*T. Laine	Cegelec
H. Landet	FINT
C. Langford	Cullen G. Langford Inc.
*R. Lasher	Bailey Controls Company
P. Leviti	Marconi Italiana Automazione

* One Vote per Company

K. Lindner
 T. Madden
 *G. Mazzola
 A. McCauley, Jr.
 F. McKenna
 R. Meeker
 D. Merriman
 *O. Mirabella
 D. Modell
 A. Mukherji
 *G. Nachev
 L. Natiello
 *L. Neitzel
 H. Newman
 *P. Noury
 *E. Pageler
 P. Patel
 *T. Phinney
 A. Poschmann
 F. Russo
 K. Schwarz
 *E. Sederlund
 Y. Shimanuki
 E. Skabowski
 J. Slavinsky
 D. Sniezek
 J. Sprague
 R. Squires
 H. Storey
 R. Szanyi
 P. Thiessmeier
 *N. Tobol
 *A. Tresset
 C. Van Haren
 *J. Weiss
 C. Williams
 G. Winchester
 *D. Wroblewski
 M. Zielinski

Endress + Hauser GmbH
 Exxon Company USA
 CISE
 Chagrin Valley Controls, Inc.
 FMcK Associates, Ltd.
 Process Control Solutions, Inc.
 MerTech, Inc.
 Universita Degli Studi di Catania
 Process & Control
 Systems Automation International Inc.
 Isomatic Laboratories
 Merck and Company Inc.
 Fisher•Rosemount Systems Inc.
 Cornell University
 Cegelec
 Fisher•Rosemount Systems Inc.
 V Automation
 Honeywell Inc.
 Ifak
 ENEL-Ricerca/PEA
 Schwarz Communication Consult
 Dow Chemical Company
 Toshiba Corporation
 Tendent Consulting
 Czech Standards Institute
 Lockheed Martin Federal Services
 Saudi Aramco Oil Company
 Measurement Technology Ltd.
 Shell Oil Company
 Mobil Tech Company
 Klockner-Moeller
 Ronan Engineering
 Cegelec
 James River Corporation
 Electric Power Research Institute
 Eastman Kodak Company
 National Electric. Manufacturers Association
 Bailey Controls Company
 Fisher•Rosemount Systems Inc.

* One Vote per Company

This Standard was approved for publication by the ISA Standards and Practices Board on September 30, 1998.

NAME

COMPANY

R. Webb, Vice President	Altran Corporation
H. Baumann	H. D. Baumann, Inc.
D. Bishop	Chevron Production Company Technology
P. Brett	Honeywell Inc.
W. Calder III	Calder Enterprises
M. Cohen	Senior Flexonics, Inc.
H. Dammeyer	The Ohio State University
W. Holland	Southern Company
H. Hopkins	Consultant
A. Iverson	Ivy Optiks
K. Lindner	Endress + Hauser GmbH
V. Maggioli	Feltronics Corporation
T. McAviney	Instrumentation and Control Engineering LLC
A. McCauley, Jr.	Chagrin Valley Controls, Inc.
G. McFarland	Honeywell Inc.
E. Montgomery	Fluor Daniel, Inc.
D. Rapley	VACO Rapley Inc.
R. Reimer	Rockwell Automation
J. Rennie	Factory Mutual Research Corp.
W. Weidman	Parsons Energy and Chemical Group
J. Weiss	Electric Power Research Institute
J. Whetstone	National Inst. of Standards & Technology
M. Widmeyer	Consultant
H. Wiegler	Canus Corporation
C. Williams	Eastman Kodak Company
G. Wood	Graeme Wood Consulting
M. Zielinski	Fisher•Rosemount Systems Inc.

Contents

Foreward	11
Introduction	13
1 Scope	13
2 Normative References	14
3 Definitions	15
3.1 Definitions from other ISO/IEC Standards	15
3.2 Definitions from ISA-S50.02-5	16
3.3 Other Definitions	16
3.4 Abbreviations and Symbols	17
3.5 Conventions	18
3.6 Conventions Used in State Machines	18
4 FAL Syntax Description	19
4.1 FAL-AR PDU Abstract Syntax 1	19
4.2 FAL-AR PDU Abstract Syntax 2	21
4.3 Abstract Syntax of PDUBody	23
4.4 Data Types	61
5 FAL Protocol State Machine	63
5.1 AP Context State Machine	65
5.2 FAL Service Protocol Machine (FSPM)	82
5.3 DLL Mapping Protocol Machine (DMPM)	88
Annex A — FAL Header (Normative)	104
Annex B — Reason Codes (Normative)	106
Annex C — Data Link Layer Service Selection (Normative).....	107
Annex D — Compact Encoding Rule (CER) [Normative].....	109
Annex E — Traditional Encoding Rule (TER) [Normative]	122
Annex F — Buffer-Oriented Encoding Rules (BER) and Messaging Encoding Rule (MER) [Normative]	132
Annex G — Queued Usertriggered Unidirectional (QUU) ARPM (Normative)	138
Annex H — Queued Usertriggered Bidirectional-Connection Oriented (QUB-CO) ARPM (Normative)	145
Annex I — Queued Usertriggered Bidirectional-Connectionless (QUB-CI) ARPM (Normative)	163
Annex J — Queued Usertriggered Bidirectional-Segmentation (QUB-Seg) ARPM (Normative)	172

Annex K — Queued Usertriggered Bidirectional-Flow Control (QUB-FC)	
ARPM (Normative)	195
Annex L — Buffered Usertriggered Bidirectional (BUB) ARPM	
(Normative)	226
Annex M — Buffered Networkscheduled Bidirectional (BNB) ARPM	
(Normative)	245
Annex N — Buffered Networkscheduled Unidirectional (BNU)	
ARPM (Normative)	274
Annex O — Buffered Networkscheduled Unidirectional-MP	
(BNU-MP) ARPM (Normative)	286
Annex P — Protocol Option 1 (Informative)	301
Annex Q — Protocol Option 2 (Informative)	303
Annex R — Protocol Option 3 (Informative)	305

Foreword

ISA-S50.02 consists of the following parts, under the general title *Fieldbus Standard for Use in Industrial Control Systems*.

- Part 1 Introductory Guide
- Part 2 Physical Layer Specification and Service Definition
- Part 3 Data Link Layer Service Definition
- Part 4 Data Link Layer Protocol Specification
- Part 5 Application Layer Service Definition
- Part 6 Application Layer Protocol Specification
- Part 7 System Management
- Part 8 Conformance Testing

Introduction

This part of ISA-S50.02 describes the Fieldbus Application Layer (FAL) protocol that defines the information interchange and the interactions between Application Entity invocations (AE-Is) to support the services defined in ISA-S50.02-5.

An Application Process uses the Fieldbus Application Layer services to exchange information with other Application Processes. The services define the abstract interface between the application process and the Application Layer.

The Application Layer protocol is the set of rules that governs the format and meaning of the information exchange between the Application Layers in various devices. The Application Layer uses the protocol to implement the Application Layer services definitions.

The protocol machine defines the various states of an Application Layer and the valid transitions between the states. It may be considered as a finite state machine. The protocol machine is described using state tables. The information is exchanged between the application process and the protocol machine through application service data units. The protocol machine exchanges information with other protocol machines through application protocol data units (APDU).

This set of Application Layer standards does not specify individual implementations or products, nor does it constrain the implementations of Application Entities (AEs) and interfaces within the industrial automation system.

This set of Application Layer standards does not contain test procedures to ensure compliance with such requirements.

1 Scope

The Fieldbus Application Layer (FAL) is an Application Layer communication standard designed to support the conveyance of time-critical application requests and responses among devices in an automation environment. The term "time-critical" is used to represent the presence of a time-window, within which one or more specified actions must be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This part of ISA-S50.02 specifies interactions between remote applications in terms of:

- The encoding rules that are applied to all the Application Layer Protocol Data Units (APDUs).
- The formal Abstract Syntax definitions of such APDUs.
- The protocol state machine descriptions that handle the APDUs and the primitives in the correct sequences.
- The mappings of the APDUs to and from the Data Link Layer services defined in ISA-S50.02-3.

The FAL encoding rules are designed assuming that both the encoder (sender) and the decoder (receiver) have the common knowledge of the abstract syntax. Wherever possible, data types identifiers are not encoded and transferred over the network.

NOTE — This is why the Abstract Syntax Notation One / Basic Encoding Rule is not practical for the FAL.

The purpose of this part of ISA-S50.02 is to define the protocol provided

- a) to the Fieldbus Data Link Layer at the boundary between the Application and Data Link Layers of the Fieldbus Reference Model, and
- b) to the System Management at the boundary between the System Management and Application Layers of the Fieldbus Reference Model.

2 Normative References

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISA-S50.02. At the time of publication, the editions were valid. All standards are subject to revision, and parties to agreements based on this part of ISA-S50.02 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC maintain registers of currently valid International Standards.

ISO/IEC 7498-1:1994, *Information processing systems – Open systems interconnection – Basic reference model: The Basic Model*

ISO/IEC 7498-2:1989, *Information processing systems – Open systems interconnection – Basic reference model Part 2 – Security architecture*

ISO/IEC 7498-3:1997, *Information technology – Open systems interconnection – Basic reference model – Naming and addressing*

ISO/IEC 8824:1990, *Information technology – Open systems interconnection – Specification of abstract syntax notation one (ASN.1)*

ISO/IEC 8825:1990, *Information Technology – Open systems interconnection – Specification of basic encoding rules for abstract syntax notation one (ASN.1)*

ANSI/ISA-S50.02, 1997, Part 3, *Fieldbus Standard for Use in Industrial Control Systems, Part 3: Data Link Service Definition*

ANSI/ISA-S50.02, 1997, Part 4, *Fieldbus Standard for Use in Industrial Control Systems, Part 4: Data Link Protocol Specification*

ANSI/ISA-S50.02, 1998, Part 5, *Fieldbus Standard for Use in Industrial Control Systems, Part 5: Application Layer Service Definition*

ISO/IEC 10646-1:1993, *Information technology – Universal Multiple – Octet coded character set (UCS) – part-1: Architecture and basic multilingual plane*

ISO/IEC 10731:1994, *Information technology – Open systems interconnection – Basic reference model – Conventions for the definition of OSI services*

ISO/IEC 8822:1994, *Information technology – Open systems interconnection – Presentation service definition*

ISO/IEC 9545:1994, *Information technology – Open systems interconnection – Application layer structure*

3 Definitions

For the purpose of ISA-S50.02-6, the following definitions apply:

3.1 Definitions from other ISO/IEC Standards

3.1.1 Definitions from ISO/IEC 7498-1

- a) application entity;
- b) application process;
- c) application protocol data unit;
- d) application service element;
- e) application entity invocation;
- f) application process invocation;
- g) application transaction;
- h) real open system;
- i) transfer syntax.

3.1.2 Definitions from ISO/IEC 8822

- a) abstract syntax;
- b) presentation context.

3.1.3 Definitions from ISO/IEC 9545

- a) application-association;
- b) application-context;
- c) application context name;
- d) application-entity-invocation;
- e) application-entity-type;
- f) application-process-invocation;
- g) application-process-type;
- h) application-service-element;
- i) application control service element.

3.1.4 Definitions from ISO/IEC 8824

- a) BOOLEAN;
- b) INTEGER;
- c) BIT STRING;
- d) OCTET STRING;
- e) NULL;
- f) SEQUENCE;
- g) SEQUENCE OF;
- h) IMPLICIT;
- i) CHOICE;
- j) ANY;
- k) OPTIONAL;
- l) DEFAULT;
- m) TRUE;
- n) FALSE.

3.1.5 Definitions from ISO/IEC 8825

- a) encoding (of a data value);
- b) data value;
- c) Identifier Octets (the singular form is used in this part of ISA-S50.02);
- d) Length Octet(s) (both singular and plural forms are used in this part of ISA-S50.02);
- e) Contents Octets.

3.2 Definitions from ISA-S50.02-5

- a) application relationship;
- b) conveyance path;
- c) client;
- d) dedicated AR;
- e) dynamic AR;
- f) error class;
- g) error code;
- h) name;
- i) numeric identifier;
- j) peer;
- k) pre-defined AR endpoint;
- l) pre-established AR endpoint;
- m) publisher;
- n) server.

3.3 Other Definitions

3.3.1 called

A service user or a service provider that receives an indication primitive or a request APDU.

3.3.2 calling

A service user or a service provider that initiates a request primitive or a request APDU.

3.3.3 interoperability

The capability of User Layer entities to perform coordinated and cooperative operations using the services of the FAL.

3.3.4 management information

Network accessible information that supports the management of the Fieldbus environment.

3.3.5 receiving

A service user that receives a confirmed primitive or an unconfirmed primitive, or a service provider that receives a confirmed APDU or an unconfirmed APDU.

3.3.6 resource

A resource is a processing or information capability of a subsystem.

3.3.7 sending

A service user that sends a confirmed primitive or an unconfirmed primitive, or a service provider that sends a confirmed APDU or an unconfirmed APDU.

3.4 Abbreviations and Symbols

AE	Application Entity
AE-I	Application Entity Invocation
AL	Application Layer
AP	Application Process
Ap_	Prefix for Data types defined for AP ASE
Ar_	Prefix for Data types defined for AR ASE
APDU	Application Protocol Data Unit
AR	Application Relationship
AREP	Application Relationship End Point
ASE	Application Service Element
ASN.1	Abstract Syntax Notation One
BCD	Binary Coded Decimal
BER	Basic Encoding Rule
cnf	confirmation primitive
DI_	Prefix for Data types defined for Data Link Layer types
DL	Data Link
DLC	Data Link Connection
DLCEP	Data Link Connection End Point
DLSAP	Data Link Service Access Point
DLSDU	Data Link Service Data Unit
Dt_	Prefix for Data types defined for Data Type ASE
Err	Error (used to indicate an APDU type)
Er_	Prefix for Error types defined
Ev_	Prefix for Data types defined for Event ASE
FAL	Fieldbus Application Layer
Fi_	Prefix for Data types defined for Function Invocation ASE
FIFO	First In First Out
Gn_	Prefix for Data Types defined for general use
ID	Identifier
IEC	International Electrotechnical Commission
in	input primitive
ind	indication primitive
ISA	Instrument Society of America
ISO	International Organization For Standardization
LAS	Link Active Scheduler
Lr_	Prefix for Data types defined for Load Region ASE
lsb	least significant bit
Mn_	Prefix for Data Types defined for Management ASE
msb	most significant bit
out	output primitive
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PICS	Protocol Implementation Conformance Statement
QoS	Quality Of Service
Req	Request (used to indicate an APDU type)
req	request primitive
Rsp	Response (used to indicate an APDU type)
rsp	response primitive
SAP	Service Access Point
SDU	Service Data Unit
ToS	Type Of Service
Vr_	Prefix for Data types defined for Variable ASE

3.5 Conventions

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate clause. Each ASE specification is composed of three parts: its class definitions, its services, and its protocol specification. The first two are contained in Part-5 of ISA-S50.02. The protocol specification for each of the ASEs is defined in this part of ISA-S50.02.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in the ISA-S50.02-5 specification. The service specification defines the services that are provided by the ASE.

3.5.1 General Conventions

This part of ISA-S50.02 uses the descriptive conventions given in ISO/IEC 10731:1994.

3.5.2 Conventions for Class Definitions

The Data Link Layer mapping definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is defined in ISA-S50.02-5.

3.5.3 Abstract Syntax Conventions

When the "optionalParametersMap" parameter is used, a bit number which corresponds to each OPTIONAL or DEFAULT production is given as a comment.

3.6 Conventions Used in State Machines

The State Machines are described as follows:

Table 1 — Conventions used for State Machines

#	Current State	Events Actions	Next State
Name of this transition.	The current state to which this state transition applies.	Events or conditions that trigger this state transaction. The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions.	The next state after the actions in this transition is taken.

The conventions used in the state machines are as follows:

:= Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.

xxx A parameter name.

Example:

Identifier := reason

means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'

"xxx" Indicates fixed value.

Example:

Identifier := "abc"

means value "abc" is assigned to a parameter named 'Identifier.'

= A logical condition to indicate an item on the left is equal to an item on the right.

< A logical condition to indicate an item on the left is less than the item on the right.

> A logical condition to indicate an item on the left is greater than the item on the right.

<> A logical condition to indicate an item on the left is not equal to an item on the right.

&& Logical "AND"

|| Logical "OR"

```
for (Identifier := start_value to end_value)
    actions
endfor
```

This construct allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

```
If (condition)
    actions
else
    actions
endif
```

This construct allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition.

Readers are strongly recommended to refer to the clauses for the AREP attribute definitions, the local functions, and the FAL-PDU definitions to understand protocol machines. It is assumed that readers have sufficient knowledge of these definitions and they are used without further explanations.

4 FAL Syntax Description

```
FalArHeader ::= Unsigned8 {
    -- bit 8      FAL Protocol Specifier      (Always 0 for the protocol defined by this part of ISA-S50.02)
    -- bit 7 - 4  Protocol Identifier          (Identifies abstract syntax, revision, and encoding rules)
    -- bit 3      Protocol Specific bit        (Reserved for each protocol to use)
    -- bit 2-1    PDU Identifier               (Identifies a PDU type within a Protocol Identifier)
}
```

The FalArHeader shall be the first octet of all the FAL PDUs. It enables multiple protocols as well as encoding rules coexist on the same network. "Annex-A FAL Header" defines the code points for the currently identified combinations of protocol and encoding rules.

4.1 FAL-AR PDU Abstract Syntax 1

The productions defined here shall be used with the Compact Encoding Rules and strongly recommended to be used with the messaging and buffered encoding rules to produce the shortest PDUs that are suitable for a limited bandwidth underlying layer.

```

FalArPDU ::= ConfirmedSend-CommandPDU
            || ConfirmedSend-AffirmativePDU
            || ConfirmedSend-NegativePDU
            || UnconfirmedSend-CommandPDU
            || UnconfirmedAcknowledgedSend-CommandPDU
            || UnconfirmedAcknowledgedSend-AffirmativePDU
            || IdleSend-CommandPDU
            || AR-XON-OFF-CommandPDU
            || Establish-CommandPDU
            || Establish-AffirmativePDU
            || Establish-NegativePDU
            || Abort-PDU

```

4.1.1 Confirmed Send Service

```

ConfirmedSend-CommandPDU ::= SEQUENCE {
    FalArHeader,
    InvokeID,
    ConfirmedServiceRequest
}

```

```

ConfirmedSend-AffirmativePDU ::= SEQUENCE {
    FalArHeader,
    InvokeID,
    ConfirmedServiceResponse
}

```

```

ConfirmedSend-NegativePDU ::= SEQUENCE {
    FalArHeader,
    InvokeID,
    ConfirmedServiceError
}

```

4.1.2 Unconfirmed Send Service

```

UnconfirmedSend-CommandPDU ::= SEQUENCE {
    FalArHeader,
    InvokeID,
    UnconfirmedServiceRequest
}

```

4.1.3 Unconfirmed Acknowledged Send Service

```

UnconfirmedAcknowledgedSend-CommandPDU ::= SEQUENCE {
    FalArHeader,
    unconfirmed-PDU [4] IMPLICIT SEQUENCE {
        InvokeID,
        UnconfirmedServiceRequest
    }
}

```

```

UnconfirmedAcknowledgedSend-AffirmativePDU ::= {
    FalArHeader
}

```

4.1.4 Idle Send Service

```

IdleSend-CommandPDU ::= {
    FalArHeader
}

```

4.1.5 AR-XON-OFF Send Service

```

AR-XON-OFF-CommandPDU ::= SEQUENCE {
    FalArHeader,
    AR-XON-OFFPDU [7] IMPLICIT SEQUENCE {
        Invoke ID,
        XON-OFF
    }
}

```

}

4.1.6 Establish Service

```
Establish-CommandPDU ::= SEQUENCE {  
    FalArHeader,  
    RespondingAREP,  
    RequestingAREP,  
    MAXOSCC,  
    MAXOSCS,  
    MAXUCSC,  
    MAXUCSS,  
    CIU,  
    InvokeID,  
    Initiate-RequestPDU  
}
```

```
Establish-AffirmativePDU ::= SEQUENCE {  
    FalArHeader,  
    InvokeID,  
    Initiate-ResponsePDU  
}
```

```
Establish-NegativePDU ::= SEQUENCE {  
    FalArHeader,  
    InvokeID,  
    Initiate-ErrorPDU  
}
```

4.1.7 Abort Service

```
Abort-PDU ::= SEQUENCE {  
    FalArHeader,  
    Identifier,  
    ReasonCode,  
    AdditionalDetail  
}
```

4.2 FAL-AR PDU Abstract Syntax 2

The productions defined here shall be used with the Traditional Encoding Rules to maintain compatibility with prior standards.

```
FalArPDU ::= ConfirmedSend-RequestPDU  
           || ConfirmedSend-ResponsePDU  
           || UnconfirmedSend-PDU  
           || UnconfirmedAcknowledgedSend-CommandPDU  
           || UnconfirmedAcknowledgedSend-AffirmativePDU  
           || IdleSend-CommandPDU  
           || AR-XON-OFF-CommandPDU  
           || Establish-RequestPDU  
           || Establish-Request2PDU  
           || Establish-ResponsePDU  
           || Establish-ErrorPDU  
           || Abort-PDU
```

4.2.1 Confirmed Send Service

```
ConfirmedSend-RequestPDU ::= SEQUENCE {  
    FalArHeader,  
    confirmed-RequestPDU  
        InvokeID,  
        ConfirmedServiceRequest  
    }  
}
```

```

ConfirmedSend-ResponsePDU ::= SEQUENCE {
    FalArHeader,
    PduBody CHOICE {
        confirmed-ResponsePDU          [2] IMPLICIT SEQUENCE {
            InvokeID,
            ConfirmedServiceResponse
        },
        confirmed-ErrorPDU              [3] IMPLICIT SEQUENCE {
            InvokeID,
            ConfirmedServiceError
        }
    }
}

```

4.2.2 Unconfirmed Send Service

```

UnconfirmedSend-PDU ::= SEQUENCE {
    FalArHeader,
    unconfirmed-PDU                    [4] IMPLICIT SEQUENCE {
        InvokeID,
        UnconfirmedServiceRequest
    }
}

```

4.2.3 Unconfirmed Acknowledged Send Service

```

UnconfirmedAcknowledgedSend-CommandPDU ::= SEQUENCE {
    FalArHeader,
    unconfirmed-PDU                    [4] IMPLICIT SEQUENCE {
        InvokeID,
        UnconfirmedServiceRequest
    }
}

```

```

UnconfirmedAcknowledgedSend-AffirmativePDU ::= {
    FalArHeader
}

```

4.2.4 Idle Send Service

```

IdleSend-CommandPDU ::= {
    FalArHeader
}

```

4.2.5 AR-XON-OFF Send Service

```

AR-XON-OFF-CommandPDU ::= SEQUENCE {
    FalArHeader,
    AR-XON-OFFPDU                      [7] IMPLICIT SEQUENCE {
        Invoke ID,
        XON-OFF
    }
}

```

4.2.6 Establish Service

```

Establish-RequestPDU ::= SEQUENCE {
    FalArHeader,
    RespondingAREP,
    RequestingAREP,
    userData                                [6] IMPLICIT SEQUENCE {
        InvokeID,
        InitiateRequest                    [0] IMPLICIT Initiate-RequestPDU
    }
}

```

```

Establish-Request2PDU ::= SEQUENCE {
    FalArHeader,
    RespondingAREP,
    RequestingAREP,
    AREPContext
    MaxOSCC,
    MaxOSCS,
    MAXUCSC,
    MAXUCSS,
    CIU
}

Establish-ResponsePDU ::= SEQUENCE {
    FalArHeader,
    userData
    InvokeID,
    InitiateResponse
}

Establish-ErrorPDU ::= SEQUENCE {
    FalArHeader,
    userData
    InvokeID,
    initiateError
}

```

[5] IMPLICIT SEQUENCE {

[6] IMPLICIT SEQUENCE {

[1] IMPLICIT Initiate-ResponsePDU

[6] IMPLICIT SEQUENCE {

[2] IMPLICIT Initiate-ErrorPDU

4.2.7 Abort Service

```

Abort-PDU ::= SEQUENCE {
    FalArHeader,
    Identifier,
    ReasonCode,
    AdditionalDetail
}

```

4.3 Abstract Syntax of PDUBody

4.3.1 InvokeID

InvokeID ::= Unsigned8

4.3.2 MaxOSCC

MaxOSCC ::= Unsigned8

4.3.3 MaxOSCS

MaxOSCS ::= Unsigned8

4.3.4 MaxUCSC

MaxUCC ::= Unsigned8

4.3.5 MaxUCSS

MaxUCS ::= Unsigned8

4.3.6 XON_OFF

XON_OFF ::= Unsigned8

4.3.7 CIU

CIU ::= Unsigned32

4.3.8 ConfirmedServiceRequest

```
ConfirmedServiceRequest ::= CHOICE {  
    status-Request  
    identify-Request  
    read2-Request  
    write-Request  
    getAttributes2-Request  
    readWithType-Request  
    writeWithType-Request  
    defineVariableList-Request  
    deleteVariableList-Request  
    initiateClientPullDownloadStatic-Request  
    pullDownloadSegment-Request  
    terminatePullDownload-Request  
    initiateClientPullUploadStatic-Request  
    pullUploadSegment-Request  
    terminatePullUpload-Request  
    initiateServerPullDownload-Request  
    initiateServerPullUpload-Request  
    createFunctionInvocation-Request  
    deleteFunctionInvocation-Request  
    start-Request  
    stop-Request  
    resume-Request  
    reset-Request  
    kill-Request  
    enableEvent-Request  
    acknowledgeEventNotification-Request  
    physicalRead-Request  
    physicalWrite-Request  
    beginSetAttributes-Request  
    setAttributes2-Request  
    endSetAttributes-Request  
    initiateClientPushDownload-Request  
    pushDownloadSegment-Request  
    terminatePushDownload-Request  
    discard-Request  
    readList-Request  
    writeList-Request  
    exchange-Request  
    getEventSummary-Request  
    getEventSummaryList-Request  
    setAttributes1-Request  
    getAttributes1-Request  
    read1-Request  
    conclude-Request  
    subscribe-Request  
    initiateClientPullDownloadDynamic-Request  
    initiateClientPullUploadDynamic-Request  
    exchangeList-Request  
    enableEventList-Request  
    confirmedAcknowledgeEventList-Request  
    getAttributes3-Request  
    queryEventSummaryList-Request  
}  
[0] IMPLICIT Status-RequestPDU,  
[1] IMPLICIT Identify-RequestPDU,  
[2] IMPLICIT Read2-RequestPDU,  
[3] IMPLICIT Write-RequestPDU,  
[4] IMPLICIT GetAttributes2-RequestPDU,  
[5] IMPLICIT ReadWithType-RequestPDU,  
[6] IMPLICIT WriteWithType-RequestPDU,  
[7] IMPLICIT DefineVariableList-RequestPDU,  
[8] IMPLICIT DeleteVariableList-RequestPDU,  
[9] IMPLICIT InitiateClientPullDownloadStatic-RequestPDU,  
[10] IMPLICIT PullDownloadSegment-RequestPDU,  
[11] IMPLICIT TerminatePullDownload-RequestPDU,  
[12] IMPLICIT InitiateClientPullUploadStatic-RequestPDU,  
[13] IMPLICIT PullUploadSegment-RequestPDU,  
[14] IMPLICIT TerminatePullUpload-RequestPDU,  
[15] IMPLICIT InitiateServerPullDownload-RequestPDU,  
[16] IMPLICIT InitiateServerPullUpload-RequestPDU,  
[17] IMPLICIT CreateFunctionInvocation-RequestPDU,  
[18] IMPLICIT DeleteFunctionInvocation-RequestPDU,  
[19] IMPLICIT Start-RequestPDU,  
[20] IMPLICIT Stop-RequestPDU,  
[21] IMPLICIT Resume-RequestPDU,  
[22] IMPLICIT Reset-RequestPDU,  
[23] IMPLICIT Kill-RequestPDU,  
[24] IMPLICIT EnableEvent-RequestPDU,  
[25] IMPLICIT AcknowledgeEventNotification-RequestPDU,  
[26] IMPLICIT PhysicalRead-RequestPDU,  
[27] IMPLICIT PhysicalWrite-RequestPDU,  
[28] IMPLICIT BeginSetAttributes-RequestPDU,  
[29] IMPLICIT SetAttributes2-RequestPDU,  
[30] IMPLICIT EndSetAttributes-RequestPDU,  
[31] IMPLICIT InitiateClientPushDownload-RequestPDU,  
[32] IMPLICIT PushDownloadSegment-RequestPDU,  
[33] IMPLICIT TerminatePushDownload-RequestPDU,  
[34] IMPLICIT Discard-RequestPDU,  
[35] IMPLICIT ReadList-RequestPDU,  
[36] IMPLICIT WriteList-RequestPDU,  
[37] IMPLICIT Exchange-RequestPDU,  
[38] IMPLICIT GetEventSummary-RequestPDU,  
[39] IMPLICIT GetEventSummaryList-RequestPDU,  
[40] IMPLICIT SetAttributes1-RequestPDU,  
[41] IMPLICIT GetAttributes1-RequestPDU,  
[42] IMPLICIT Read1-RequestPDU,  
[43] IMPLICIT Conclude-RequestPDU,  
[44] IMPLICIT Subscribe-RequestPDU,  
[45] IMPLICIT InitiateClientPullDownloadDynamic-RequestPDU,  
[46] IMPLICIT InitiateClientPullUploadDynamic-RequestPDU,  
[47] IMPLICIT ExchangeList-RequestPDU,  
[48] IMPLICIT EnableEventList-RequestPDU,  
[49] IMPLICIT ConfirmedAcknowledgeEventList-RequestPDU,  
[50] IMPLICIT GetAttributes3-RequestPDU,  
[51] IMPLICIT QueryEventSummaryList-RequestPDU
```

4.3.9 ConfirmedServiceResponse

```
ConfirmedServiceResponse ::= CHOICE {  
    status-Response  
    identify-Response  
    read2-Response  
    write-Response  
    getAttributes2-Response  
    readWithType-Response  
    writeWithType-Response  
    defineVariableList-Response  
    deleteVariableList-Response  
    initiateClientPullDownloadStatic-Response  
    pullDownloadSegment-Response  
    terminatePullDownload-Response  
    initiateClientPullUploadStatic-Response  
    pullUploadSegment-Response  
    terminatePullUpload-Response  
    initiateServerPullDownload-Response  
    initiateServerPullUpload-Response  
    createFunctionInvocation-Response  
    deleteFunctionInvocation-Response  
    start-Response  
    stop-Response  
    resume-Response  
    reset-Response  
    kill-Response  
    enableEvent-Response  
    acknowledgeEventNotification-Response  
    physicalRead-Response  
    physicalWrite-Response  
    beginSetAttributes-Response  
    setAttributes2-Response  
    endSetAttributes-Response  
    initiateClientPushDownload-Response  
    pushDownloadSegment-Response  
    terminatePushDownload-Response  
    discard-Response  
    readList-Response  
    writeList-Response  
    exchange-Response  
    getEventSummary-Response  
    getEventSummaryList-Response  
    setAttributes1-Response  
    getAttributes1-Response  
    read1-Response  
    conclude-Response  
    subscribe-Response  
    initiateClientPullDownloadDynamic-Response  
    initiateClientPullUploadDynamic-Response  
    exchangeList-Response  
    enableEventList-Response  
    confirmedAcknowledgeEventList-Response  
    getAttributes3-Response  
    queryEventSummaryList-Response  
    [0] IMPLICIT Status-ResponsePDU,  
    [1] IMPLICIT Identify-ResponsePDU,  
    [2] IMPLICIT Read2-ResponsePDU,  
    [3] IMPLICIT Write-ResponsePDU,  
    [4] IMPLICIT GetAttributes2-ResponsePDU,  
    [5] IMPLICIT ReadWithType-ResponsePDU,  
    [6] IMPLICIT WriteWithType-ResponsePDU,  
    [7] IMPLICIT DefineVariableList-ResponsePDU,  
    [8] IMPLICIT DeleteVariableList-ResponsePDU,  
    [9] IMPLICIT InitiateClientPullDownloadStatic-ResponsePDU,  
    [10] IMPLICIT PullDownloadSegment-ResponsePDU,  
    [11] IMPLICIT TerminatePullDownload-ResponsePDU,  
    [12] IMPLICIT InitiateClientPullUploadStatic-ResponsePDU,  
    [13] IMPLICIT PullUploadSegment-ResponsePDU,  
    [14] IMPLICIT TerminatePullUpload-ResponsePDU,  
    [15] IMPLICIT InitiateServerPullDownload-ResponsePDU,  
    [16] IMPLICIT InitiateServerPullUpload-ResponsePDU,  
    [17] IMPLICIT CreateFunctionInvocation-ResponsePDU,  
    [18] IMPLICIT DeleteFunctionInvocation-ResponsePDU,  
    [19] IMPLICIT Start-ResponsePDU,  
    [20] IMPLICIT Stop-ResponsePDU,  
    [21] IMPLICIT Resume-ResponsePDU,  
    [22] IMPLICIT Reset-ResponsePDU,  
    [23] IMPLICIT Kill-ResponsePDU,  
    [24] IMPLICIT EnableEvent-ResponsePDU,  
    [25] IMPLICIT AcknowledgeEventNotification-ResponsePDU,  
    [26] IMPLICIT PhysicalRead-ResponsePDU,  
    [27] IMPLICIT PhysicalWrite-ResponsePDU,  
    [28] IMPLICIT BeginSetAttributes-ResponsePDU,  
    [29] IMPLICIT SetAttributes2-ResponsePDU,  
    [30] IMPLICIT EndSetAttributes-ResponsePDU,  
    [31] IMPLICIT InitiateClientPushDownload-ResponsePDU,  
    [32] IMPLICIT PushDownloadSegment-ResponsePDU,  
    [33] IMPLICIT TerminatePushDownload-ResponsePDU,  
    [34] IMPLICIT Discard-ResponsePDU,  
    [35] IMPLICIT ReadList-ResponsePDU,  
    [36] IMPLICIT WriteList-ResponsePDU,  
    [37] IMPLICIT Exchange-ResponsePDU,  
    [38] IMPLICIT GetEventSummary-ResponsePDU,  
    [39] IMPLICIT GetEventSummaryList-ResponsePDU,  
    [40] IMPLICIT SetAttributes1-ResponsePDU,  
    [41] IMPLICIT GetAttributes1-ResponsePDU,  
    [42] IMPLICIT Read1-ResponsePDU,  
    [43] IMPLICIT Conclude-ResponsePDU,  
    [44] IMPLICIT Subscribe-ResponsePDU,  
    [45] IMPLICIT InitiateClientPullDownloadDynamic-ResponsePDU,  
    [46] IMPLICIT InitiateClientPullUploadDynamic-ResponsePDU,  
    [47] IMPLICIT ExchangeList-ResponsePDU,  
    [48] IMPLICIT EnableEventList-ResponsePDU,  
    [49] IMPLICIT ConfirmedAcknowledgeEventList-ResponsePDU,  
    [50] IMPLICIT GetAttributes3-ResponsePDU,  
    [51] IMPLICIT QueryEventSummaryList-ResponsePDU  
}
```

4.3.10 ConfirmedServiceError

```
ConfirmedServiceError ::= CHOICE {  
    status-Error [0] IMPLICIT ErrorType,  
    identify-Error [1] IMPLICIT ErrorType,  
    read2-Error [2] IMPLICIT ErrorType,  
    write-Error [3] IMPLICIT ErrorType,  
    getAttributes2-Error [4] IMPLICIT ErrorType,  
    readWithType-Error [5] IMPLICIT ErrorType,  
    writeWithType-Error [6] IMPLICIT ErrorType,  
    defineVariableList-Error [7] IMPLICIT ErrorType,  
    deleteVariableList-Error [8] IMPLICIT ErrorType,  
    initiateClientPullDownloadStatic-Error [9] IMPLICIT ErrorType,  
    pullDownloadSegment-Error [10] IMPLICIT ErrorType,  
    terminatePullDownload-Error [11] IMPLICIT ErrorType,  
    initiateClientPullUploadStatic-Error [12] IMPLICIT ErrorType,  
    pullUploadSegment-Error [13] IMPLICIT ErrorType,  
    terminatePullUpload-Error [14] IMPLICIT ErrorType,  
    initiateServerPullDownload-Error [15] IMPLICIT ErrorType,  
    initiateServerPullUpload-Error [16] IMPLICIT ErrorType,  
    createFunctionInvocation-Error [17] IMPLICIT ErrorType,  
    deleteFunctionInvocation-Error [18] IMPLICIT ErrorType,  
    start-Error [19] IMPLICIT ErrorType,  
    stop-Error [20] IMPLICIT ErrorType,  
    resume-Error [21] IMPLICIT ErrorType,  
    reset-Error [22] IMPLICIT ErrorType,  
    kill-Error [23] IMPLICIT ErrorType,  
    enableEvent-Error [24] IMPLICIT ErrorType,  
    acknowledgeEventNotification-Error [25] IMPLICIT ErrorType,  
    physicalRead-Error [26] IMPLICIT ErrorType,  
    physicalWrite-Error [27] IMPLICIT ErrorType,  
    beginSetAttributes-Error [28] IMPLICIT ErrorType,  
    setAttributes2-Error [29] IMPLICIT ErrorType,  
    endSetAttributes-Error [30] IMPLICIT ErrorType,  
    initiateClientPushDownload-Error [31] IMPLICIT ErrorType,  
    pushDownloadSegment-Error [32] IMPLICIT ErrorType,  
    terminatePushDownload-Error [33] IMPLICIT ErrorType,  
    discard-Error [34] IMPLICIT ErrorType,  
    readList-Error [35] IMPLICIT ErrorType,  
    writeList-Error [36] IMPLICIT ErrorType,  
    exchange-Error [37] IMPLICIT ErrorType,  
    getEventSummary-Error [38] IMPLICIT ErrorType,  
    getEventSummaryList-Error [39] IMPLICIT ErrorType,  
    setAttributes1-Error [40] IMPLICIT ErrorType,  
    getAttributes1-Error [41] IMPLICIT ErrorType,  
    read1-Error [42] IMPLICIT ErrorType,  
    conclude-Error [43] IMPLICIT ErrorType,  
    subscribe-Error [44] IMPLICIT ErrorType,  
    initiateClientPullDownloadDynamic-Error [45] IMPLICIT ErrorType,  
    initiateClientPullDownloadDynamic-Error [46] IMPLICIT ErrorType,  
    exchangeList-Error [47] IMPLICIT ErrorType,  
    enableEventList-Error [48] IMPLICIT ErrorType,  
    confirmedAcknowledgeEventList-Error [49] IMPLICIT ErrorType,  
    getAttributes3-Error [50] IMPLICIT ErrorType,  
    queryEventSummaryList-Error [51] IMPLICIT ErrorType,  
    xon-off-Error [52] IMPLICIT ErrorType  
}
```

4.3.11 Error Type

```
ErrorType ::= SEQUENCE {  
    errorClass  
    optionalParametersMap  
    additionalCode  
    additionalDescription  
    serviceSpecificInfo  
}
```

[0] IMPLICIT ErrorClass,
[10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
[1] IMPLICIT Integer16 OPTIONAL,
[2] IMPLICIT VisibleString OPTIONAL,
[3] IMPLICIT ANY OPTIONAL

4.3.12 Error Class

```
ErrorClass ::= CHOICE {  
    vfdState  
        other  
    },  
    applicationReference  
        other  
        application-unreachable  
        application-reference-invalid  
        context-unsupported  
    },  
    definition  
        other  
        object-undefined  
        object-attributes-inconsistent  
        name-already-exists  
        type-unsupported  
        type-inconsistent  
    },  
    resource  
        other  
        memory-unavailable  
    },  
    service  
    access  
    objectDescription  
        other  
        name-length-overflow  
        od-overflow  
        od-write-protected  
        extension-length-overflow  
        od-description-length-overflow  
        operational-problem  
    },  
    conclude  
        other  
        further-communication-required  
    },  
    other  
}
```

[1] IMPLICIT Integer8 {
 (0)
}
[2] IMPLICIT Integer8 {
 (0),
 (1),
 (2),
 (3)
}
[3] IMPLICIT Integer8 {
 (0),
 (1),
 (2),
 (3),
 (4),
 (5)
}
[4] IMPLICIT Integer8 {
 (0),
 (1)
}
[5] IMPLICIT Er_Service,
[6] IMPLICIT Er_Access,
[7] IMPLICIT Integer8 {
 (0),
 (1),
 (2),
 (3),
 (4),
 (5),
 (6)
}
[9] IMPLICIT Integer8 {
 (0),
 (1)
}
[8] IMPLICIT Er_Other

4.3.12.1 Er_Service

```
Er_Service ::= Integer8 {  
    other  
    object-state-conflict  
    pdu-size  
    object-constraint-conflict  
    parameter-inconsistent  
    illegal-parameter  
}
```

(0),
(1),
(2),
(3),
(4),
(5)

4.3.12.2 Er_Access

```
Er_Access ::= Integer8 {  
    other (0),  
    object-invalidated (1),  
    hardware-fault (2),  
    object-access-denied (3),  
    invalid-address (4),  
    object-attribute-inconsistent (5),  
    object-access-unsupported (6),  
    object-non-existent (7),  
    type-conflict (8),  
    named-access-unsupported (9),  
    access-to-element-unsupported (10)  
},
```

4.3.12.3 Er_Other

```
Er-Other ::= Integer8 {  
    other (0)  
}
```

4.3.13 Unconfirmed PDUs

```
UnconfirmedServiceRequest ::= CHOICE {  
    informationReport-Request [0] IMPLICIT InformationReport-RequestPDU,  
    unsolicitedStatus-Request [1] IMPLICIT UnsolicitedStatus-RequestPDU,  
    eventNotification-Request [2] IMPLICIT EventNotification-RequestPDU,  
    informationReportWithType [3] IMPLICIT InformationReportWithType-RequestPDU,  
    eventNotificationWithType [4] IMPLICIT EventNotificationWithType-RequestPDU,  
    actionInvoke-Request [5] IMPLICIT ActionInvoke-RequestPDU,  
    actionReturn-Request [6] IMPLICIT ActionReturn-RequestPDU,  
    informationReportList-Request [7] IMPLICIT InformationReportList-RequestPDU,  
    reject-Request [8] IMPLICIT Reject-RequestPDU,  
    notificationRecovery-Request [9] IMPLICIT NotificationRecovery-RequestPDU,  
    eventNotificationList-Request [10] IMPLICIT EventNotificationList-RequestPDU  
}
```

4.3.14 Management ASE

4.3.14.1 Begin Set Attributes Service

```
BeginSetAttributes-RequestPDU ::= Integer8 {  
    loadingNonInteracting (0),  
    appendingInteracting (1),  
    freshLoadingInteracting (2)  
}
```

BeginSetAttributes -ResponsePDU ::= NULL

4.3.14.2 Create Service

4.3.14.2.1 Create Function Invocation PDU

```
CreateFunctionInvocation-RequestPDU ::= SEQUENCE {
    listOfRelatedObjects      [0] IMPLICIT SEQUENCE OF Gn_AccessSpecification,
    accessPrivilege           [1] IMPLICIT Fi_AccessPrivilege,
    optionalParametersMap     [10] IMPLICIT Gn_OptionalParametersMap16 OPTIONAL,
    fiName                    [2] IMPLICIT Gn_Name OPTIONAL,           -- bit 1
    fiIndex                   [12] IMPLICIT Gn_NumericID OPTIONAL,      -- bit 2
    userDescription           [3] IMPLICIT ANY OPTIONAL,                -- bit 3
    reusable                  [4] IMPLICIT Gn_Reusable DEFAULT TRUE,     -- bit 4
    deletable                 [5] IMPLICIT Gn_Deletable OPTIONAL,        -- bit 5
    startServiceArgumentType  [6] IMPLICIT Gn_NumericID OPTIONAL,       -- bit 6
    stopServiceArgumentType   [7] IMPLICIT Gn_NumericID OPTIONAL,       -- bit 7
    resumeServiceArgumentType [8] IMPLICIT Gn_NumericID OPTIONAL,       -- bit 8
    resetServiceArgumentType  [9] IMPLICIT Gn_NumericID OPTIONAL,       -- bit 9
    killServiceArgumentType   [11] IMPLICIT Gn_NumericID OPTIONAL       -- bit 10
}
```

CreateFunctionInvocation-ResponsePDU ::= Gn_NumericID

4.3.14.2.2 Define Variable List PDU

```
DefineVariableList-RequestPDU ::= SEQUENCE {
    listOfVariables      [0] IMPLICIT SEQUENCE OF Gn_AccessSpecification,
    accessPrivilege      [1] IMPLICIT Vr_AccessPrivilege,
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    variableListName     [2] IMPLICIT Gn_Name OPTIONAL,           -- bit 1
    variableListIndex    [5] IMPLICIT Gn_NumericID OPTIONAL,      -- bit 2
    userDescription      [3] IMPLICIT ANY OPTIONAL,                -- bit 3
    deletable            [4] IMPLICIT Gn_Deletable OPTIONAL       -- bit 4
}
```

DefineVariableList-ResponsePDU ::= Gn_NumericID

4.3.14.3 Delete Service

4.3.14.3.1 Delete Function Invocation PDU

DeleteFunctionInvocation-RequestPDU ::= Gn_AccessSpecification

DeleteFunctionInvocation-ResponsePDU ::= NULL

4.3.14.3.2 Delete Variable List PDU

DeleteVariableList-RequestPDU ::= Gn_AccessSpecification

DeleteVariableList-ResponsePDU ::= NULL

4.3.14.4 End Set Attributes Service

EndSetAttributes-RequestPDU ::= NULL

EndSetAttributes-ResponsePDU ::= NULL

4.3.14.5 Get Attributes List Service

```
GetAttributes1-RequestPDU ::= CHOICE {
    listOfObjectsAndAttributes [0] IMPLICIT SEQUENCE OF SEQUENCE {
        falClass      [6] IMPLICIT Gn_NumericID,
        keyAttribute   Gn_KeyAttribute,
        listOfRequestedAttributes [7] IMPLICIT Mn_AttributesMap
    },
    rangeOfObjectsAndAttributes [1] IMPLICIT SEQUENCE {
        startingNumericId [0] IMPLICIT Gn_NumericID,
        listOfRequestedAttributes [1] IMPLICIT Mn_AttributesMap
    }
}
```

```

GetAttributes2-RequestPDU ::= SEQUENCE {
    listOfAttributes
    accessSpecification CHOICE {
        numericID [0] IMPLICIT Mn_SelectedAttributes,
        variableName [1] IMPLICIT Gn_NumericID,
        variableListName [2] IMPLICIT Gn_Name,
        domainName [3] IMPLICIT Gn_Name,
        fiName [4] IMPLICIT Gn_Name,
        eventName [5] IMPLICIT Gn_Name,
        startIndex [6] IMPLICIT Gn_Name,
        [7] IMPLICIT Gn_NumericID
    }
}

GetAttributes3-RequestPDU ::= SEQUENCE {
    objectClass [2] IMPLICIT Gn_ObjectClass,
    start objectID Gn_AccessSpecification
}

GetAttributes1-ResponsePDU ::= SEQUENCE {
    listOfAttributeIdAndValues [0] IMPLICIT Mn_AttributeValues,
    more [1] IMPLICIT Gn_MoreFollows
}

GetAttributes2-ResponsePDU ::= SEQUENCE {
    listOfObjectDefinition [0] IMPLICIT SEQUENCE OF Gn_ObjectDefinition,
    more [1] IMPLICIT Gn_MoreFollows
}

GetAttributes3-ResponsePDU ::= CHOICE {
    listofobjectID [0] IMPLICIT SEQUENCE {
        listofID [2] IMPLICIT SEQUENCE OF Gn_AccessSpecification,
        morefollows [3] IMPLICIT Gn_MoreFollows
    }, --The object class value in the request is <> 0
    listofobjectDefinition [1] IMPLICIT SEQUENCE {
        listofDefinition [4] IMPLICIT SEQUENCE OF Mn_AttributeValues
        morefollows [5] IMPLICIT Gn-MoreFollows
    } --The object class value in the request is = 0
}

```

4.3.14.6 Set Attributes Service

```

SetAttributes1-RequestPDU ::= SEQUENCE OF SEQUENCE {
    falClass [6] IMPLICIT Gn_NumericID,
    keyAttributes Gn_KeyAttributes,
    listOfAttributeUpdates [7] IMPLICIT Mn_AttributeValues
}

SetAttributes2-RequestPDU ::= SEQUENCE OF {
    objectDefinition [0] IMPLICIT Gn_ObjectDefinition
}

SetAttributes1-ResponsePDU ::= NULL

SetAttributes2-ResponsePDU ::= NULL

```

4.3.15 Application Process ASE

4.3.15.1 Get Status Service

Status-RequestPDU ::= NULL

```
Status-ResponsePDU ::= SEQUENCE {
    logicalStatus
        state-changes-allowed          [0] IMPLICIT Unsigned8 {
        no-state-changes-allowed        (0),
        limited-services-permitted      (1),
        od-loading-non-interacting      (2),
        od-loading-interacting          (4),
        },
        physicalStatus
        operational                     [1] IMPLICIT Unsigned8 {
        partially-operational            (0),
        inoperable                      (1),
        needs-commissioning              (2),
        },
    optionalParametersMap               [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    localDetail                         [2] IMPLICIT BitString OPTIONAL
}
-- bit 1
```

4.3.15.2 Identify Service

Identify-RequestPDU ::= NULL

```
Identify-ResponsePDU ::= SEQUENCE {
    vendorName                         [0] IMPLICIT VisibleString,
    modelIdentifier                    [1] IMPLICIT VisibleString,
    vendorRevision                     [2] IMPLICIT VisibleString,
    optionalParametersMap              [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    serialNumber                       [3] IMPLICIT OctetString OPTIONAL
}
-- bit 1
```

4.3.15.3 Initiate Service

```
Initiate-RequestPDU ::= SEQUENCE {
    versionObjectDefinitionsCalling    [0] IMPLICIT Integer16,
    apDescriptorCalling                [1] IMPLICIT OctetString,
    accessProtectionSupportedCalling   [2] IMPLICIT Ap_AccessProtectionSupported,
    passwordAndAccessGroupsCalling     [3] IMPLICIT Ap_AccessControl,
    configuredMaxPduSizeSending        [4] IMPLICIT Unsigned8,
    configuredMaxPduSizeReceiving      [5] IMPLICIT Unsigned8,
    listOfSupportedServicesCalling      [6] IMPLICIT Mn_PduSupportedMap,
    configuredMaxOutstandingServicesRequesting [7] IMPLICIT Unsigned8,
    configuredMaxOutstandingServicesResponding [8] IMPLICIT Unsigned8,
    optionalParametersMap              [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    userDetail                         [11] IMPLICIT OctetString OPTIONAL,
    configuredNestingLevel             [12] IMPLICIT Unsigned8 DEFAULT 1
}
-- bit 1
-- bit 2
```

```
Initiate-ResponsePDU ::= SEQUENCE {
    versionObjectDefinitionsCalled     [0] IMPLICIT Integer16,
    profileNumberCalled                [1] IMPLICIT OctetString,
    accessPrivilegeSupportedCalled     [2] IMPLICIT Ap_AccessProtectionSupported,
    passwordAndAccessGroupsCalled      [3] IMPLICIT Ap_AccessControl,
    negotiatedMaxOutstandingServicesRequesting [4] IMPLICIT Unsigned8,
    negotiatedMaxOutstandingServicesResponding [5] IMPLICIT Unsigned8,
    optionalParametersMap              [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    userDetail                         [11] IMPLICIT OctetString OPTIONAL,
    negotiatedNestingLevel             [12] IMPLICIT Unsigned8 DEFAULT 1,
    negotiatedListOfServices           [13] IMPLICIT Mn_PduSupportedMap OPTIONAL
}
-- bit 1
-- bit 2
-- bit 3
```

```

Initiate-ErrorPDU ::= SEQUENCE {
    errorCode
        other
        max-fal-pdu-size-insufficient
        service-not-supported
        version-obj-def-incompatible
        user-initiate-denied
        password-error
        profile-number-incompatible
    },
    maxPduLengthSendingCalled
    maxPduLengthReceivingCalled
    listOfSupportedServicesCalled
}

[0] IMPLICIT Integer8 {
    (0),
    (1),
    (2),
    (3),
    (4),
    (5),
    (6)
}

[1] IMPLICIT Unsigned8,
[2] IMPLICIT Unsigned8,
[3] IMPLICIT Mn_PduSupportedMap
-- See NOTE
-- See NOTE

```

NOTE — For QUB-Seg the PDU length is given in multiples of 256 octets, i.e. value 4 means 4*256 = 1024 octets

4.3.15.4 Status Notification Service

UnsolicitedStatus-RequestPDU ::= Status-ResponsePDU

4.3.15.5 Subscribe Service

```

Subscribe-RequestPDU ::= CHOICE {
    joining
        publishedObject
        scheduleInterval
        optionalParametersMap
        maximumARs
        maximumPermissibleJitter
        desiredPublishingStartTime
        earliestPublishingStartTime
        latestPublishingStartTime
    },
    listOfLeavingAREPs
}

[0] IMPLICIT SEQUENCE {
    Gn_KeyAttributes,
    [11] IMPLICIT DI_CyclicInterval,
    [12] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    [13] IMPLICIT Unsigned8 DEFAULT 1 OPTIONAL,
    [14] IMPLICIT DI_PermissibleJitter OPTIONAL,
    [15] IMPLICIT FieldbusTime OPTIONAL,
    [16] IMPLICIT FieldbusTime OPTIONAL,
    [17] IMPLICIT FieldbusTime OPTIONAL
    -- bit 1
    -- bit 2
    -- bit 3
    -- bit 4
    -- bit 5
}

[1] IMPLICIT SEQUENCE OF Gn_NumericID

Subscribe-ResponsePDU ::= SEQUENCE {
    optionalParametersMap
    dataType
    dataLength
    dedicated
    listOfArInformation
        publishingAREPid
        publishingDIMapping
        scheduledStartTime
    } OPTIONAL
}

[0] Gn_OptionalParametersMap8 OPTIONAL,
[1] Gn_NumericID OPTIONAL,
[2] DI_SduSize OPTIONAL,
[3] Ar_Dedicated DEFAULT TRUE,
[4] IMPLICIT SEQUENCE OF SEQUENCE {
    [0] Gn_NumericID,
    [1] Gn_NumericID,
    [2] FieldbusTime
    -- bit 1
    -- bit 2
    -- bit 3
    -- bit 4
}

```

4.3.15.6 Reject Service

```

RejectPDU ::= SEQUENCE {
    original-invokeID
    reject-code
        other
        illegal-confirmed-service-request
        pdu-error
        pdu-size
    }
}

[0] IMPLICIT Integer8,
[1] IMPLICIT Integer8 {
    (0),
    (1),
    (2),
    (5)
}

```

4.3.15.7 Conclude Service

Conclude-Request ::= NULL

Conclude-Response ::= NULL

4.3.16 Load Region ASE

4.3.16.1 Discard Service

Discard-RequestPDU ::= Gn_KeyAttribute -- Load Region LocalID

Discard-ResponsePDU ::= NULL

4.3.16.2 Initiate Load Service

4.3.16.2.1 InitiateClientPullDownloadStatic PDUs

InitiateClientPullDownloadStatic-RequestPDU ::= Gn_AccessSpecification

InitiateClientPullDownloadStatic-ResponsePDU ::= NULL

4.3.16.2.2 InitiateClientPullDownloadDynamic PDUs

```
InitiateClientPullDownloadDynamic-RequestPDU ::= SEQUENCE {
    loadRegionKeyAttribute      Gn_AccessSpecification,
    userData                    [0] OctetString,
    sharable                    [1] Boolean,
    accessProtection            [2] Fi_AccessPrivilege
}
```

InitiateClientPullDownloadDynamic-ResponsePDU ::= NULL

4.3.16.2.3 InitiateClientPullUploadStatic PDUs

InitiateClientPullUploadStatic-RequestPDU ::= Gn_AccessSpecification

InitiateClientPullUploadStatic-ResponsePDU ::= NULL

4.3.16.2.4 InitiateClientPullUploadDynamic PDUs

InitiateClientPullUploadDynamic-RequestPDU ::= Gn_AccessSpecification

```
InitiateClientPullUploadDynamic-ResponsePDU ::= SEQUENCE {
    ulsmNumericID              [0] Gn_NumericID,
    userData                   [1] OctetString
}
```

4.3.16.2.5 InitiateServerPullDownload PDUs

```
InitiateServerPullDownload-RequestPDU ::= SEQUENCE {
    loadRegionKeyAttribute      Gn_AccessSpecification,
    optionalParametersMap      [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    additionalInformation       [11] IMPLICIT VisibleString OPTIONAL
} -- bit 1
```

InitiateServerPullDownload-ResponsePDU ::= NULL

4.3.16.2.6 InitiateServerPullUpload PDUs

```
InitiateServerPullUpload-RequestPDU ::= SEQUENCE {
    loadRegionKeyAttribute      Gn_AccessSpecification,
    optionalParametersMap      [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    additionalInformation       [2] IMPLICIT VisibleString OPTIONAL
} -- bit 1
```

InitiateServerPullUpload-ResponsePDU ::= NULL

4.3.16.2.7 InitiateClientPushDownload PDUs

InitiateClientPushDownload-RequestPDU ::= Gn_AccessSpecification

InitiateClientPushDownload-ResponsePDU ::= NULL

4.3.16.3 Pull Segment Service

4.3.16.3.1 PullDownloadSegment PDUs

```
PullDownloadSegment-RequestPDU ::= SEQUENCE {  
    loadRegionKeyAttribute      Gn_AccessSpecification  
    optionalParametersMap      [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    segmentNumber              [11] IMPLICIT Fi_SegmentNumber OPTIONAL  
}
```

-- bit 1

```
PullDownloadSegment-ResponsePDU ::= SEQUENCE {  
    loadData                   [0] IMPLICIT OctetString,  
    moreFollows                [1] IMPLICIT Gn_MoreFollows  
}
```

4.3.16.3.2 PullUploadSegment PDUs

PullUploadSegment-RequestPDU ::= Gn_AccessSpecification

```
PullUploadSegment-ResponsePDU ::= SEQUENCE {  
    loadData                   [0] IMPLICIT OctetString,  
    moreFollows                [1] IMPLICIT Gn_MoreFollows  
}
```

4.3.16.4 Push Segment Service

4.3.16.4.1 PushDownloadSegment PDUs

```
PushDownloadSegment-RequestPDU ::= SEQUENCE {  
    loadRegionKeyAttribute      Gn_AccessSpecification,  
    loadData                   [2] IMPLICIT OctetString,  
    moreFollows                [3] IMPLICIT Gn_MoreFollows,  
    optionalParametersMap      [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    segmentNumber              [11] IMPLICIT Fi_SegmentNumber OPTIONAL  
}
```

-- bit 1

PushDownloadSegment-ResponsePDU ::= NULL

4.3.16.5 Terminate Load Service

4.3.16.5.1 TerminatePullDownload PDUs

```
TerminatePullDownload-RequestPDU ::= SEQUENCE {  
    loadRegionKeyAttribute      Gn_AccessSpecification,  
    terminateReason            [2] IMPLICIT Lr_TerminateReason  
}
```

TerminatePullDownload-ResponsePDU ::= NULL

4.3.16.5.2 TerminatePushDownload PDUs

TerminatePushDownload-RequestPDU ::= Gn_AccessSpecification

TerminatePushDownload-ResponsePDU ::= Lr_TerminateReason

4.3.16.5.3 TerminatePullUpload PDUs

TerminatePullUpload-RequestPDU ::= Gn_AccessSpecification

TerminatePullUpload-ResponsePDU ::= NULL

4.3.17 Function Invocation ASE

4.3.17.1 ActionInvoke Service

```
ActionInvoke-RequestPDU ::= SEQUENCE {  
    keyAttribute          Gn_KeyAttribute,  
    actionInvocationID    [6] IMPLICIT Fi_ActionInvokeID,  
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    executionArgument     [7] IMPLICIT OctetString OPTIONAL  
}
```

-- bit 1

4.3.17.2 ActionReturn Service

```
ActionReturn-RequestPDU ::= SEQUENCE {  
    actionInvocationID [0] IMPLICIT Fi_ActionInvokeID,  
    actionResults CHOICE {  
        actionSuccess [1] IMPLICIT ANY,  
        actionFailure [2] IMPLICIT ErrorType  
    }  
}
```

4.3.17.3 Kill Service

```
Kill-RequestPDU ::= SEQUENCE {  
    keyAttribute          Gn_KeyAttribute,  
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    executionArgument     [7] IMPLICIT OctetString OPTIONAL  
}
```

-- bit 1

Kill-ResponsePDU ::= NULL

4.3.17.4 Reset Service

```
Reset-RequestPDU ::= SEQUENCE {  
    keyAttribute          Gn_KeyAttribute,  
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    executionArgument     [7] IMPLICIT OctetString OPTIONAL  
}
```

-- bit 1

Reset-ResponsePDU ::= NULL

4.3.17.5 Resume Service

```
Resume-RequestPDU ::= SEQUENCE {  
    keyAttribute          Gn_KeyAttribute,  
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    executionArgument     [7] IMPLICIT OctetString OPTIONAL  
}
```

-- bit 1

Resume-ResponsePDU ::= NULL

4.3.17.6 Start Service

```
Start-RequestPDU ::= SEQUENCE {  
    keyAttribute          Gn_KeyAttribute,  
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    executionArgument     [7] IMPLICIT OctetString OPTIONAL  
}
```

-- bit 1

Start-ResponsePDU ::= NULL

4.3.17.7 Stop Service

```
Stop-RequestPDU ::= SEQUENCE {
    keyAttribute          Gn_KeyAttribute,
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    executionArgument     [7] IMPLICIT OctetString OPTIONAL
}
-- bit 1
```

Stop-ResponsePDU ::= NULL

4.3.18 Variable Access ASE

4.3.18.1 Exchange Service

```
Exchange-RequestPDU ::= SEQUENCE {
    specifierForVariableToWrite [0] Gn_KeyAttribute,
    valueToWrite                [1] IMPLICIT ANY,
    specifierForVariableToRead  [2] Gn_KeyAttribute,
    optionalParametersMap      [2] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    objectRevisionForVariableWriting [3] IMPLICIT Gn_ObjectRevision OPTIONAL,
    objectRevisionRequested     [4] IMPLICIT Gn_RequestFlag DEFAULT FALSE
}
-- bit 1
-- bit 2
```

```
Exchange-ResponsePDU ::= SEQUENCE {
    writeResult
    readResult CHOICE {
        status
        value
        valueWithRevision
        value
        objectRevision
    }
}
[0] IMPLICIT Vr_WriteResult,
[1] IMPLICIT Er_Access,
[2] IMPLICIT ANY,
[3] IMPLICIT SEQUENCE {
    [4] IMPLICIT ANY,
    [5] IMPLICIT Gn_ObjectRevision
}
```

4.3.18.2 Exchange List Service

```
ExchangeList-RequestPDU ::= SEQUENCE {
    writeList-Request
    readList-Request
}
[0] IMPLICIT WriteList-RequestPDU,
[1] IMPLICIT ReadList-RequestPDU
```

```
ExchangeList-Response ::= SEQUENCE {
    writeList-Response
    readList-Response
}
[0] IMPLICIT WriteList-ResponsePDU,
[1] IMPLICIT ReadList-ResponsePDU
```

4.3.18.3 Information Report Service

```
InformationReport-RequestPDU ::= SEQUENCE {
    value [4] IMPLICIT ANY,
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    variableSpecifier Gn_KeyAttribute OPTIONAL,
    subIndex [3] IMPLICIT Gn_SubIndex OPTIONAL,
    objectRevision [11] IMPLICIT Gn_ObjectRevision OPTIONAL
}
-- bit 1
-- bit 2
-- bit 3
```

4.3.18.4 Information Report with Type Service

```
InformationReportWithType-RequestPDU ::= SEQUENCE {
    TypeDescription [4] Gn_TypeDescription,
    value [5] IMPLICIT ANY,
    optionalParametersMap [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    variableSpecifier Gn_KeyAttribute OPTIONAL,
    subIndex [3] IMPLICIT Gn_SubIndex OPTIONAL,
    objectRevision [11] IMPLICIT Gn_ObjectRevision OPTIONAL
}
-- bit 1
-- bit 2
-- bit 3
```

4.3.18.5 Information List Report Service

```
InformationReportList-RequestPDU ::= SEQUENCE {
    ListOfVariableSpecifier CHOICE {
        variableList          [0] IMPLICIT Gn_KeyAttribute,
        listOfVariable        [1] IMPLICIT Gn_ListOfVariableAccess
    },
    listOfDataType            [7] IMPLICIT SEQUENCE OF Gn_FullyNestedTypeDescription,
    listOfData                [8] IMPLICIT SEQUENCE OF ANY,
    optionalParametersMap     [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    userDetails               [11] IMPLICIT OctetString OPTIONAL,          -- bit 1
    listForObjectRevision     [12] SEQUENCE OF Gn_ObjectRevision OPTIONAL  -- bit 2
}

-- The listOfDataType should either contain one element per variable or should be empty. The listForObjectRevision
-- should contain one element per variable or should be absent.
```

4.3.18.6 Read Service

4.3.18.6.1 Read PDUs

```
Read1-RequestPDU ::= SEQUENCE {
    variableSpecifier          Gn_KeyAttribute,
    optionalParametersMap     [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    objectRevisionRequested   [11] IMPLICIT Gn_RequestFlag DEFAULT FALSE OPTIONAL  -- bit 1
}

Read2-RequestPDU ::= SEQUENCE {
    variableSpecifier          Gn_KeyAttribute,
    optionalParametersMap     [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    subIndex                  [3] IMPLICIT Gn_SubIndex OPTIONAL                  -- bit 1
}

Read1-ResponsePDU ::= SEQUENCE {
    value                     [0] IMPLICIT ANY,
    optionalParametersMap     [1] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    objectRevision            [2] IMPLICIT Gn_ObjectRevision OPTIONAL              -- bit 1
}

Read2-ResponsePDU ::= ANY
```

4.3.18.6.2 PhysicalRead PDUs

```
PhysicalRead-RequestPDU ::= SEQUENCE {
    local-address             [0] IMPLICIT Gn_PhysicalAddress,
    length                    [1] IMPLICIT Gn_Length
}

PhysicalRead-ResponsePDU ::= ANY
```

4.3.18.6.3 ReadWithType PDUs

```
ReadWithType-RequestPDU ::= SEQUENCE{
    variableSpecifier          Gn_KeyAttribute,
    optionalParametersMap     [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    subIndex                  [3] IMPLICIT Gn_SubIndex OPTIONAL                  -- bit 1
}

ReadWithType-ResponsePDU ::= SEQUENCE{
    typeDescription           [0] Gn_TypeDescription,
    value                     [1] IMPLICIT ANY,
    optionalParametersMap     [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    objectRevision            [11] IMPLICIT Gn_ObjectRevision OPTIONAL              -- bit 1
}
```

4.3.18.7 Read List Service

```
ReadList-RequestPDU ::= SEQUENCE {
    ListOfVariableSpecifier CHOICE {
        variableList
        listOfVariable
        listOfNumericAddress
    },
    additionalInfoRequested BitString8 {
        typeDescriptionRequested
        objectRevisionRequested
    }
}

ReadList-ResponsePDU ::= SEQUENCE {
    ListOfAccessResult
    listOfDataType
    listOfData
    listOfObjectRevision
}
```

[0] IMPLICIT Gn_KeyAttributes,
[1] IMPLICIT Gn_ListOfVariableAccess,
[2] IMPLICIT SEQUENCE OF Gn_NumericAddress

(0),
(1)

[0] IMPLICIT BitString,
[1] IMPLICIT SEQUENCE OF Gn_FullyNestedTypeDescription,
[2] IMPLICIT SEQUENCE OF ANY,
[3] IMPLICIT SEQUENCE OF Gn_ObjectRevision

- The listOfAccessResult parameter specifies for each requested variable, if the access was successful (bit = 1) or not (bit = 0).
- The listOfDataType should either contain one element per variable or should be empty. The same applies for the listOfObjectRevision.
- The listOfData parameter specifies for each requested variable the value of the variable if the access succeeds or an ErrorClass if it fails. The ErrorClass is coded as an unsigned16 where the most significant byte contains the "error class" and the least significant byte contains the "error code" (see [ErrorClass](#) definition).

4.3.18.7.1 Write Service

4.3.18.7.2 Write PDUs

```
Write-RequestPDU ::= SEQUENCE {
    variableSpecifier
    value
    optionalParametersMap
    subIndex
    objectRevision
}
```

Gn_KeyAttribute,
[4] IMPLICIT ANY,
[10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
[3] IMPLICIT Gn_SubIndex OPTIONAL, -- bit 1
[11] IMPLICIT Gn_ObjectRevision OPTIONAL -- bit 2

Write-ResponsePDU ::= NULL

4.3.18.7.3 PhysicalWrite PDUs

```
PhysicalWrite-RequestPDU ::= SEQUENCE {
    local-address
    data
}
```

[0] IMPLICIT Gn_PhysicalAddress,
[1] IMPLICIT ANY

PhysicalWrite-ResponsePDU ::= NULL

4.3.18.7.4 WriteWithType PDUs

```
WriteWithType-RequestPDU ::= SEQUENCE{
    VariableSpecifier
    typeDescription
    value
    optionalParametersMap
    subIndex
    objectRevision
}
```

Gn_KeyAttribute,
[4] Gn_TypeDescription,
[5] IMPLICIT ANY,
[10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
[3] IMPLICIT Gn_SubIndex OPTIONAL, -- bit 1
[11] IMPLICIT Gn_ObjectRevision OPTIONAL -- bit 2

WriteWithType-ResponsePDU ::= NULL

4.3.18.8 Write List Service

```
WriteList-RequestPDU ::= SEQUENCE {
    ListOfVariableSpecifier CHOICE {
        variableList          [0] IMPLICIT Gn_KeyAttributes,
        listOfVariable        [1] IMPLICIT Gn_ListOfVariableAccess,
        listOfNumericAddress  [2] IMPLICIT SEQUENCE OF Gn_NumericAddress
    },
    listOfDataType            [3] IMPLICIT SEQUENCE OF Gn_FullyNestedTypeDescription,
    listOfValue               [4] IMPLICIT SEQUENCE OF ANY,
    listOfObjectRevision      [5] IMPLICIT SEQUENCE OF Gn_ObjectRevision
}
```

-- The listOfDataType should either contain one element per variable or should be empty. The same applies for the listOfObjectRevision.

```
WriteList-ResponsePDU ::= SEQUENCE {
    ListOfAccessResult        [0] IMPLICIT BitString,
    listOfAccessError         [1] IMPLICIT SEQUENCE OF ErrorClass
}
```

-- The listOfAccessResult parameter specifies for each requested variable, if the access was successful (bit = 1) or not (bit = 0).

-- The listOfAccessError contains one error code for each failed access. If a variable access was successful, there is no corresponding error code in this list.

-- The ErrorClass is coded as an unsigned16 where the most significant byte contains the "error class" and the least significant byte contains the "error code" (see [ErrorClass](#) definition).

4.3.19 Event Management ASE

4.3.19.1 Confirmed Acknowledge Event Service

```
AcknowledgeEventNotification-RequestPDU ::= SEQUENCE {
    keyAttribute              Gn_AccessSpecification,
    reportedEventCount        [2] IMPLICIT Ev_EventCount
}
```

AcknowledgeEventNotification-ResponsePDU ::= NULL

4.3.19.2 EnableEvent Service

```
EnableEvent-RequestPDU ::= SEQUENCE {
    keyAttribute              Gn_AccessSpecification,
    enabled                   [2] IMPLICIT Gn_Enabled
}
```

EnableEvent-ResponsePDU ::= NULL

4.3.19.3 EnableEventList Service

```
EnableEventList-RequestPDU ::= SEQUENCE {
    listOfKeyAttribute        [0] IMPLICIT SEQUENCE OF Gn_AccessSpecification,
    listOfEnabledFlags        [1] IMPLICIT BitString
}
```

```
EnableEventList-ResponsePDU ::= SEQUENCE {
    listOfEnableResult        [0] IMPLICIT BitString,
    listOfError               [1] IMPLICIT ErrorClass
}
```

4.3.19.4 Event Notification Service

```
EventNotificationList-RequestPDU ::= SEQUENCE {
    eventNotifierID          IMPLICIT Gn_NumericID,
    eventDataContained        Boolean,
    notificationTypeInfo CHOICE {
        basic                [0] NULL,
        sequenced            [1] Ev_SequenceNumber,
        time-of-notification [2] Ev_TimeTag,
        compound             [3] IMPLICIT SEQUENCE {
            sequence-number  Ev_SequenceNumber,
            time-tag         Ev_TimeTag
        }
    },
    listOfEventID            SEQUENCE OF Gn_AccessSpecification,
    listOfEv_EventDataTypes SEQUENCE OF Gn_FullyNestedTypeDescription,
    listOfEventContents CHOICE {
        simple               [0] SEQUENCE OF ANY, -- NULL, Ev_EventData,
        reportedEventCount   [1] SEQUENCE OF ANY, -- Ev_EventCount, Ev_EventData
        eventDetectionTime   [2] SEQUENCE OF ANY, -- Ev_TimeTag, Ev_EventData
        composite            [3] SEQUENCE OF ANY -- Ev_EventCount, Ev_TimeTag,
                                                Ev_EventData
    }
}
```

-- If the eventDataContained parameter is TRUE, the Ev_EventData value shall be present.

-- The components of event contents shall be appended together without any gap.

```
EventNotification-RequestPDU ::= SEQUENCE {
    KeyAttribute             Gn_KeyAttribute,
    eventNumber              [2] IMPLICIT Unsigned8,
    optionalParametersMap    [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    eventData                [3] IMPLICIT ANY OPTIONAL, -- bit 1
    eventNotifierID          [11] IMPLICIT Gn_NumericID OPTIONAL, -- bit 2
    sequenceNumber           [12] IMPLICIT Ev_SequenceNumber OPTIONAL, -- bit 3
    notificationTime         [13] IMPLICIT Ev_TimeTag OPTIONAL -- bit 4
}
```

4.3.19.5 Event Notification with Type Service

```
EventNotificationWithType-RequestPDU ::= SEQUENCE{
    KeyAttribute             Gn_KeyAttribute,
    eventNumber              [2] IMPLICIT Unsigned8,
    typeDescription          [3] Gn_TypeDescription,
    optionalParametersMap    [10] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    value                   [4] IMPLICIT ANY OPTIONAL, -- bit 1
    eventNotifierID          [11] IMPLICIT Gn_NumericID OPTIONAL, -- bit 2
    sequenceNumber           [12] IMPLICIT Ev_SequenceNumber OPTIONAL, -- bit 3
    notificationTime         [13] IMPLICIT Ev_TimeTag OPTIONAL -- bit 4
}
```

4.3.19.6 Get Event Summary Service

```
GetEventSummary-RequestPDU ::= SEQUENCE {
    eventID                 [0] IMPLICIT Gn_NumericID,
    optionalParametersMap    [1] IMPLICIT OptionalParametersMap8 OPTIONAL,
    recoveredMessageRequested [2] IMPLICIT Gn_RequestFlag OPTIONAL, -- bit 1
    objectRevisionRequested  [3] IMPLICIT Gn_RequestFlag OPTIONAL -- bit 2
}
```

```
GetEventSummary-ResponsePDU ::= SEQUENCE {
    eventStatus             [0] IMPLICIT Ev_EventStatus,
    optionalParametersMap    [1] IMPLICIT OptionalParametersMap8 OPTIONAL,
    objectRevision          [2] IMPLICIT Gn_ObjectRevision OPTIONAL, -- bit 1
    recoveredMessage        [3] IMPLICIT Ev_EventMessage OPTIONAL -- bit 2
}
```

4.3.19.7 Get Event Summary List Service

```
GetEventSummaryList-RequestPDU ::= SEQUENCE {
    request events CHOICE {
        allEvents [1] IMPLICIT NULL,
        listOfEvents [2] IMPLICIT SEQUENCE OF Gn_AccessSpecification
    },
    additionalInfoRequest [3] IMPLICIT BitString8 {
        recoveredMessageRequested (0), --true for requested
        objectRevisionRequested (1) --true for requested
    }
}

GetEventSummaryList-ResponsePDU ::= CHOICE {
    listOfEvents [1] IMPLICIT SEQUENCE {
        listOfAccessResult [0] IMPLICIT BitString,
        listOfEventMessage Ev_ListOfEventResponse,
        listOfObjectRevision [5] IMPLICIT SEQUENCE OF Gn_ObjectRevision
    },
    allEvents [2] IMPLICIT SEQUENCE {
        listOfEventID [0] IMPLICIT SEQUENCE OF Gn_AccessSpecification,
        listOfEventMessage Ev_ListOfEventResponse,
        listOfObjectRevision [5] IMPLICIT SEQUENCE OF Gn_ObjectRevision
    }
}

-- The listOfAccessResult parameter specifies for each requested event, if the access was successful (bit=1) or not (bit=0).
```

4.3.19.8 Notification Recovery Service

```
NotificationRecovery-RequestPDU ::= SEQUENCE {
    notifierID [0] IMPLICIT Gn_NumericID,
    optionalParametersMap [1] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
    numberToRecover [2] IMPLICIT Ev_NumberToRecover DEFAULT 1 OPTIONAL, -- bit 1
    sequenceNumber [3] IMPLICIT Ev_SequenceNumber OPTIONAL -- bit 2
}
```

4.3.19.9 Confirmed Acknowledge Event List Service

```
ConfirmedAcknowledgedEventList-RequestPDU ::= SEQUENCE {
    acknowledgmentPolicy [0] IMPLICIT Ev_AcknowledgmentPolicy,
    listOfEventID [1] IMPLICIT SEQUENCE OF Gn_AccessSpecification,
    listOfMessageSpecifier CHOICE {
        reportedEventCount [2] IMPLICIT SEQUENCE OF Ev_EventCount,
        eventDetectionTime [3] IMPLICIT SEQUENCE OF Ev_TimeTag,
        eventCountAndDetectionTime [4] IMPLICIT SEQUENCE OF ANY -- Ev_EventCount, Ev_TimeTag
    },
    listOfAcknowledgmentData [5] IMPLICIT SEQUENCE OF OctetString
}

ConfirmedAcknowledgedEventList-ResponsePDU ::= SEQUENCE {
    listOfAccessResult [0] IMPLICIT BitString,
    listOfError [1] IMPLICIT ErrorClass
}
```

4.3.19.10 Query Event Summary List Service

```
QueryEventSummaryList-RequestPDU ::= SEQUENCE {
    eventselectionCriterion [0] IMPLICIT BitString8,
    requestedEvent CHOICE {
        allEvents [1] IMPLICIT NULL,
        listOfEvents [2] IMPLICIT SEQUENCE OF Gn_AccessSpecification
    },
    additionalInfoRequested [3] IMPLICIT BitString8 {
        recoveredMessageRequested (0), --true for requested
        objectRevisionRequested (1) --true for requested
    }
}
```


4.3.20.2.3.2 Reason Code

ReasonCode ::= Unsigned8

4.3.20.2.3.3 Additional Detail

AdditionalDetail ::= OctetString

4.3.20.2.4 Ar_ServiceTimeOut

Ar_ServiceTimeOut ::= Unsigned8

4.3.20.3 Data Link Layer Types

4.3.20.3.1 DI_CyclicInterval

DI_CyclicInterval ::= Unsigned32 -- Granularity is defined in ISA-S50.02-3 and ISA-S50.02-4.

4.3.20.3.2 DI_DlsapAddress

DI_DlsapAddress ::= Unsigned32 -- The format of this type is defined in ISA-S50.02-4.

4.3.20.3.3 DI_PermissibleJitter

DI_PermissibleJitter ::= Unsigned16 -- Granularity is defined in ISA-S50.02-3 and ISA-S50.02-4.

4.3.20.3.4 DI_SduSize

DI_SduSize ::= Unsigned16

4.3.20.4 Data Type ASE Types

4.3.20.4.1 Dt_ListOfFields

Dt_ListOfFields ::= SEQUENCE {
 fieldDataType [0] IMPLICIT Gn_NumericID,
 optionalParametersMap [2] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,
 fieldName [0] IMPLICIT Gn_Name OPTIONAL
}

-- bit 1

4.3.20.5 Event ASE Types

4.3.20.5.1 Ev_AcknowledgmentDataSpecifier

Ev_AcknowledgmentDataSpecifier ::= Ev_DataSpecifier

4.3.20.5.2 Ev_AcknowledgmentPolicy

Ev_AcknowledgmentPolicy ::= Boolean -- True means all messages are to be acknowledged.
-- False means only the specified message is to be acknowledged.

4.3.20.5.3 Ev_ContainedEventData

Ev_ContainedEventData ::= Boolean -- True means the event message contains event data.
-- False means the event message does not contain event data.

4.3.20.5.4 Ev_ContainedMessageType

Ev_ContainedMessageType ::= Unsigned8 {
 simple (1),
 reportedEventCount (2),
 eventDetectionTime (3),
 composite (4),
 any (5)
}

4.3.20.5.5 Ev_DataSpecifier

```
Ev_DataSpecifier ::= CHOICE {  
    variableID  
    dataTypeId  
    dataLength  
}  
[0] IMPLICIT Gn_NumericID,  
[1] IMPLICIT Gn_NumericID,  
[2] IMPLICIT Unsigned8
```

4.3.20.5.6 Ev_EventCount

Ev_EventCount ::= Unsigned8

4.3.20.5.7 Ev_EventData

Ev_EventData ::= ANY

-- Ev_EventData contains the data associated with the respective event. If the data is not to be included in the notification, its length is 0.

4.3.20.5.8 Ev_EventMessage

```
Ev_EventMessage ::= CHOICE {  
    eventKeyAttributeOnly  
    eventKeyAttributeWithCount  
    eventKeyAttributeWithTime  
    eventKeyAttributeWithData  
}  
[0] IMPLICIT Gn_KeyAttribute,  
[1] IMPLICIT Ev_KeyAttributeWithMessageCount,  
[2] IMPLICIT Ev_KeyAttributeWithData,  
[3] IMPLICIT Ev_KeyAttributeWithData  
  
Ev_ListOfEventResponse ::= CHOICE {  
    simple  
    recoveredCount  
    recoveredDetectionTime  
    composite  
}  
[1] IMPLICIT SEQUENCE OF Ev_Status,  
[2] IMPLICIT SEQUENCE OF ANY, -- Ev_Status, RecoveredCount  
[3] IMPLICIT SEQUENCE OF ANY, -- Ev_Status, Recovered  
    DetectionTime  
[4] IMPLICIT SEQUENCE OF ANY -- Ev_Status, RecoveredCount,  
    RecoveredDetectionTime
```

-- The components of Ev_Status & RecoveredCount shall be appended together without any gap.

-- The components of Ev_Status & RecoveredDetectionTime shall be appended together without any gap.

-- The components of Ev_Status & RecoveredCount & RecoveredDetectionTime shall be appended together without any gap.

```
Ev_Status ::= BitString8 {  
    active  
    detected  
    enabled  
    acknowledged  
}  
(0), --true for active  
(1), --true for detected  
(2), --true for enabled  
(3) --true for acked
```

RecoveredCount ::= Unsigned8

RecoveredDetectionTime ::= Ev_TimeTag

4.3.20.5.9 Ev_EventStatus

```
Ev_EventStatus ::= SEQUENCE {  
    active  
    detected  
    enabled  
    acknowledgmentSupported  
    acknowledged  
}  
[0] IMPLICIT Boolean, -- True means active  
[1] IMPLICIT Boolean, -- True means detected  
[2] IMPLICIT Boolean, -- True means enabled  
[3] IMPLICIT Boolean, -- True means supported  
[4] IMPLICIT Boolean -- True means acknowledged
```

4.3.20.5.10 Ev_ErrorStatus

Ev_ErrorStatus ::= Boolean

-- True means the underlying condition is active.
-- False means the underlying condition is not active.

4.3.20.5.11 Ev_LastReportedEventInfo

```
Ev_LastReportedEventInfo ::= SEQUENCE {  
    optionalParametersMap      [0] IMPLICIT Gn_OptionalParametersMap8 OPTIONAL,  
    count                      [1] IMPLICIT Unsigned8 OPTIONAL,           -- bit 1  
    detectionTime              [2] IMPLICIT Ev_TimeTag OPTIONAL           -- bit 2  
}
```

4.3.20.5.12 Ev_MessageType

```
Ev_MessageType ::= Unsigned8 {  
    simpleMessage              (1),  
    reportedEventCount        (2),  
    eventDetectionTime        (3),  
    composite                  (4),  
    simpleMessageWithData     (5),  
    reportedEventCountWithData (6),  
    eventDetectionTimeWithData (7),  
    compositeWithData         (8)  
}
```

4.3.20.5.13 Ev_NotificationType

```
Ev_NotificationType ::= Unsigned8 {  
    basic                      (1),  
    sequenced                  (2),  
    timeOfNotification         (3),  
    compound                    (4)  
}
```

4.3.20.5.14 Ev_NumberToRecover

```
Ev_NumberToRecover ::= Unsigned8
```

4.3.20.5.15 Ev_SequenceNumber

```
Ev_SequenceNumber ::= Unsigned8
```

4.3.20.5.16 Ev_SelectionCriterion

```
Ev_SelectionCriterion ::= SEQUENCE {  
    enabled                    [0] IMPLICIT Boolean,  
    detected                   [1] IMPLICIT Boolean,  
    acked                      [2] IMPLICIT Boolean,  
    active                     [3] IMPLICIT Boolean  
}
```

4.3.20.5.17 Ev_TimeTag

```
Ev_TimeTag ::= Unsigned16
```

4.3.20.6 Function Invocation ASE Types

4.3.20.6.1 Fi_ActionInvokeID

```
Fi_ActionInvokeID ::= Unsigned8
```

4.3.20.6.2 Fi_AccessPrivilege

```
Fi_AccessPrivilege ::= BitString {
    rightToStartPassword          (23),
    rightToStopPassword           (22),
    rightToDeletePassword         (21),
    rightToStartAccessGroup       (19),
    rightToStopAccessGroup        (18),
    rightToDeleteAccessGroup      (17),
    rightToStartAllPartner        (31),
    rightToStopAllPartner         (30),
    rightToDeleteAllPartner       (29),
    password_Bit1                 (7),    -- The Password (Unsigned8) is encoded as a bit string.
    password_Bit2                 (6),
    password_Bit3                 (5),
    password_Bit4                 (4),
    password_Bit5                 (3),
    password_Bit6                 (2),
    password_Bit7                 (1),
    password_Bit8                 (0),
    access_Groups-1               (15),
    access_Groups-2               (14),
    access_Groups-3               (13),
    access_Groups-4               (12),
    access_Groups-5               (11),
    access_Groups-6               (10),
    access_Groups-7               (9),
    access_Groups-8               (8)
}
```

4.3.20.6.3 Fi_SegmentNumber

Fi_SegmentNumber ::= Unsigned32 -- The first segment number shall be one(1).

4.3.20.6.4 Fi_State

```
Fi_State ::= Unsigned8 {
    unrunnable                    (1),
    idle                          (2),
    running                       (3),
    stopped                       (4),
    starting                      (5),
    stopping                      (6),
    resuming                      (7),
    resetting                     (8)
}
```

4.3.20.7 General Types

4.3.20.7.1 Gn_AccessSpecification

```
Gn_AccessSpecification ::= CHOICE {
    index
    name                    [0] IMPLICIT Unsigned16,
                           [1] IMPLICIT VisibleString
}
```

4.3.20.7.2 Gn_Deletable

Gn_Deletable ::= Boolean
-- True means deletable.
-- False means not deletable.

4.3.20.7.3 Gn_Enabled

Gn_Enabled ::= Boolean
-- True means enabled.
-- False means disabled.

4.3.20.7.4 Gn_FullyNestedTypeDescription

```
Gn_FullyNestedTypeDescription ::= CHOICE {
    Boolean [1] Gn_Length,
    integer8 [2] Gn_Length,
    integer16 [3] Gn_Length,
    integer32 [4] Gn_Length,
    unsigned8 [5] Gn_Length,
    unsigned16 [6] Gn_Length,
    unsigned32 [7] Gn_Length,
    float32 [8] Gn_Length,
    float64 [15] Gn_Length,
    binaryDate [11] Gn_Length,
    timeOfDay [12] Gn_Length,
    timeDifference [13] Gn_Length,
    universalTime [16] Gn_Length,
    fieldbusTime [17] Gn_Length,
    time [21] Gn_Length,
    bitstring8 [22] Gn_Length,
    bitstring16 [23] Gn_Length,
    bitstring32 [24] Gn_Length,
    visiblestring1 [25] Gn_Length,
    visiblestring2 [26] Gn_Length,
    visiblestring4 [27] Gn_Length,
    visiblestring8 [28] Gn_Length,
    visiblestring16 [29] Gn_Length,
    octetstring1 [30] Gn_Length,
    octetstring2 [31] Gn_Length,
    octetstring4 [32] Gn_Length,
    octetstring8 [33] Gn_Length,
    octetstring16 [34] Gn_Length,
    bcd [35] Gn_Length,
    iso10646char [36] Gn_Length,
    binarytime0 [40] Gn_Length,
    binarytime1 [41] Gn_Length,
    binarytime2 [42] Gn_Length,
    binarytime3 [43] Gn_Length,
    binarytime4 [44] Gn_Length,
    binarytime5 [45] Gn_Length,
    binarytime6 [46] Gn_Length,
    binarytime7 [47] Gn_Length,
    binarytime8 [48] Gn_Length,
    binarytime9 [49] Gn_Length,
    visiblestring [9] Gn_Length,
    octetstring [10] Gn_Length,
    bitstring [14] Gn_Length,
    compactBooleanArray [37] Gn_Length,
    compactBCDArray [38] Gn_Length,
    iso646string [39] Gn_Length,
    array [55] IMPLICIT SEQUENCE {
        number-of-elements [0] IMPLICIT Unsigned8,
        element-type [1] Gn_FullyNestedTypeDescription
    },
    structure [56] IMPLICIT SEQUENCE OF Gn_FullyNestedTypeDescription
}
```

4.3.20.7.5 Gn_KeyAttribute

```
Gn_KeyAttribute ::= CHOICE {
    -- When this type is specified, only the key attributes of the class referenced are valid.
    numericID [0] IMPLICIT Gn_NumericID,
    name [1] IMPLICIT Gn_Name,
    listName [2] IMPLICIT Gn_Name,
    numericAddress [4] IMPLICIT Gn_NumericAddress,
    symbolicAddress [5] IMPLICIT Gn_SymbolicAddress
}
```

4.3.20.7.6 Gn_Length

Gn_Length ::= Unsigned8

4.3.20.7.7 Gn_ListOfVariableAccess

```
Gn_ListOfVariableAccess ::= SEQUENCE OF CHOICE {  
    numericID          [0] IMPLICIT Gn_NumericID,  
    name               [1] IMPLICIT Gn_Name,  
    listName           [2] IMPLICIT Gn_Name,  
    symbolicAddress    [5] IMPLICIT Gn_SymbolicAddress,  
    partialNumericID   [6] IMPLICIT SEQUENCE {  
        numericID      [0] IMPLICIT Gn_NumericID,  
        partialAccess  
    },  
    partialName        [7] IMPLICIT SEQUENCE {  
        name           [0] IMPLICIT Gn_Name,  
        partialAccess  
    },  
    partialListName    [8] IMPLICIT SEQUENCE {  
        listName       [0] IMPLICIT Gn_Name,  
        partialAccess  
    }  
}
```

4.3.20.7.8 Gn_MoreFollows

Gn_MoreFollows ::= Boolean

4.3.20.7.9 Gn_Name

Gn_Name ::= OctetString

4.3.20.7.10 Gn_NameWithComponent

```
Gn_NameWithComponent ::= SEQUENCE {  
    name               [0] IMPLICIT Gn_Name,  
    component          [1] IMPLICIT Unsigned16  
}
```

4.3.20.7.11 Gn_NumericAddress

```
Gn_NumericAddress ::= SEQUENCE {  
    startAddress       [0] IMPLICIT Unsigned32, --physical address of the starting location  
    length             [1] IMPLICIT Unsigned16 -- octet length of a memory block  
}
```

4.3.20.7.12 Gn_NumericID

Gn_NumericID ::= Unsigned16 -- The values of this parameter are unique within an AP.

4.3.20.7.13 Gn_NumericIdWithComponent

```
Gn_NumericIdWithComponent ::= SEQUENCE {  
    numericID          [0] IMPLICIT Gn_NumericID,  
    component          [1] IMPLICIT Unsigned16  
}
```

4.3.20.7.14 Gn_ObjectClass

```
Gn_ObjectClass ::= Unsigned8 {  
    anyObject                (0),  
    loadRegion               (2),  
    functionInvocation       (3),  
    event                    (4),  
    fixedLength&StringDataType (5),  
    structuredDataType       (6),  
    fixedLength&StringVariable (7),  
    arrayVariable            (8),  
    dataStructureVariable    (9),  
    variableList             (10),  
    arrayDataType            (12),  
    eventList                (17),  
    notifier                 (18),  
    action                   (19)  
}
```

4.3.20.7.15 Gn_ObjectDefinition

Gn_ObjectDefinition ::= OctetString -- Semantics of this parameter are application specific.

4.3.20.7.16 Gn_ObjectRevision

```
Gn_ObjectRevision ::= Unsigned8  
-- bits 8 - 5 are used as the Major Revision  
-- bits 4 - 1 are used as the Minor Revision  
-- each may have a value between zero and fifteen inclusive
```

4.3.20.7.17 Gn_OptionalParametersMaps

Gn_OptionalParametersMap8 ::= BitString8

Gn_OptionalParametersMap16 ::= BitString16

Gn_OptionalParametersMap32 ::= BitString32

- The above types have special semantics to represent which optional parameters are included
- in the transfer syntax. Bit 1 corresponds to the first optional parameter of a given primitive,
- bit 2 corresponds to the second optional parameter, up to bit "n" corresponds to the "n-th"
- optional parameters. The value of one indicates that the corresponding optional parameter is
- present in the transfer syntax, and the value of zero indicates that it is not present.

4.3.20.7.18 Gn_PartialAccess

```
Gn_PartialAccess ::= CHOICE {  
    componentID                [3] IMPLICIT Gn_SubIndex,  
    componentList              [20] IMPLICIT SEQUENCE {  
        nestingLevel          [1] Unsigned8,  
        accessedElements      [2] SEQUENCE OF Gn_Sub-Index  
    }  
}
```

- The componentID of the partial access selects a single element on the first level of nesting for access. The componentList specifies one possibly complex base element on a specified level of nesting and if necessary selects a number of subelements from this base element. The first elements (their number is given by nestingLevel) of the accessedElements list define the path to the base element. The rest of the list selects a number of subelements to be accessed. If the rest of the list is empty, the complete base element is accessed.

4.3.20.7.19 Gn_PhysicalAddress

Gn_PhysicalAddress ::= Unsigned32

4.3.20.7.20 Gn_RequestFlag

Gn_RequestFlag ::= Boolean

-- TRUE means the specified parameter is requested in the response.
-- FALSE means the specified parameter is not requested in the response.

4.3.20.7.21 Gn_Reusable

Gn_Reusable ::= Boolean

-- True means reusable.
-- False means not reusable.

4.3.20.7.22 Gn_RomRamFlag

Gn_RomRamFlag ::= Boolean

-- True means a read-only-memory is used.
-- False means a random access memory is used.

4.3.20.7.23 Gn_SubIndex

Gn_SubIndex ::= Unsigned8

4.3.20.7.24 Gn_SymbolicAddress

Gn_SymbolicAddress ::= VisibleString

4.3.20.7.25 Gn_TypeDescription

Gn_TypeDescription ::= SEQUENCE OF CHOICE {
 simple [1] IMPLICIT ANY,
 -- Octet1: High byte Data Type Gn_NumericID
 -- Octet2: Low byte Data Type Gn_NumericID
 -- Octet3: Length
 array [2] IMPLICIT ANY,
 -- Octet1: High byte Data Type Gn_NumericID
 -- Octet2: Low byte Data Type Gn_NumericID
 -- Octet3: Length
 -- Octet4: Number of Elements
 structure [3] IMPLICIT ANY
 -- Octet 1 : High byte Data Type Index (Element 1)
 -- Octet 2 : Low byte Data Type Index (Element 1)
 -- Octet 3 : Length (Element 1)
 -- Octet 3n-2 : High byte Data Type Index (Element n)
 -- Octet 3n-1 : Low byte Data Type Index (Element n)
 -- Octet 3n-0 : Length (Element n)
}

4.3.20.8 Load Region ASE Types

4.3.20.8.1 Lr_FixedLengthSegment

Lr_FixedLengthSegment ::= Boolean

-- True means LR supports fixed-length segments only.
-- False means LR supports variable-length segments.

4.3.20.8.2 Lr_LocalAddress

Lr_LocalAddress ::= Unsigned32

4.3.20.8.3 Lr_Size

Lr_Size ::= Unsigned32