

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 5-14: Application layer service definition – Type 14 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-14: Définition des services de la couche application – Éléments
de Type 14**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2010 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester.

If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de la CEI ou du Comité national de la CEI du pays du demandeur.

Si vous avez des questions sur le copyright de la CEI ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de la CEI de votre pays de résidence.

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

Useful links:

IEC publications search - www.iec.ch/searchpub

The advanced search enables you to find IEC publications by a variety of criteria (reference number, text, technical committee,...).

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available on-line and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary (IEV) on-line.

Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de la CEI

La Commission Electrotechnique Internationale (CEI) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications CEI

Le contenu technique des publications de la CEI est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Liens utiles:

Recherche de publications CEI - www.iec.ch/searchpub

La recherche avancée vous permet de trouver des publications CEI en utilisant différents critères (numéro de référence, texte, comité d'études,...).

Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

Just Published CEI - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications de la CEI. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne au monde de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans les langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (VEI) en ligne.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 5-14: Application layer service definition – Type 14 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-14: Définition des services de la couche application – Éléments
de Type 14**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XD

ICS 25.04.40; 35.100.70; 35.110

ISBN 978-2-88912-862-4

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
1.1 Overview.....	7
1.2 Specifications.....	8
1.3 Conformance.....	8
2 Normative references	8
3 Terms, definitions, symbols, abbreviations and conventions	9
3.1 ISO/IEC 7498-1 terms	9
3.2 ISO/IEC 8822 terms	9
3.3 ISO/IEC 9545 terms	9
3.4 ISO/IEC 8824-1 terms	9
3.5 Fieldbus application-layer specific definitions	9
3.6 Abbreviations and symbols	12
3.7 Conventions	13
4 Concepts.....	16
5 Data type ASE.....	16
5.1 Overview.....	16
5.2 Formal definition of data type objects.....	16
5.3 FAL defined data types.....	17
5.4 Data type ASE service specification.....	38
6 Communication model specification.....	39
6.1 General.....	39
6.2 ASEs.....	39
6.3 Application relationship.....	88
6.4 Summary of application layer services.....	91
Bibliography.....	92
Figure 1 – Application layer entity.....	39
Figure 2 – Received message processing procedure	64
Figure 3 – AR ASE conveys APDUs between AP	88
Table 1 – Attribute of variable normalised 2 octet	25
Table 2 – Encoding of variable normalised 2 octet	25
Table 3 – Attribute of normalised 4 Octet.....	26
Table 4 – Encoding of normalised 4 Octet.....	26
Table 5 – Attribute of variable normalised 2 octet	26
Table 6 – Encoding of variable normalised 2 octet	27
Table 7 – Attribute of variable normalised 4 Octet.....	27
Table 8 – Encoding of variable normalised 4 Octet	27
Table 9 – Attribute of unipolar 2 octet	28
Table 10 – Encoding of unipolar 2 octet	28
Table 11 – Attribute of Fixed point value 2 Octet.....	28

Table 12 – Encoding of Fixed point value 2 Octet	29
Table 13 – Attribute of Fixed point value 4 Octet.....	29
Table 14 – Encoding of Bit sequence 2 Octet.....	29
Table 15 – Encoding of Nibble 4 Octet	30
Table 16 – Attribute of multiple time constant 2 octets	35
Table 17 – Attribute of multiple time constant 4 octets	35
Table 18 – Attribute of fraction time constant 2 octets	36
Table 19 – Encoding of reciprocal time constant 2 octets	36
Table 20 – Management object base.....	42
Table 21 – Access group assignment.....	56
Table 22 – Access rights assignment.....	57
Table 23 – Services for domain object	57
Table 24 – Service for report object	59
Table 25 – FAL management entity services.....	66
Table 26 – EM_DetectingDevice service parameters.....	67
Table 27 – EM_OnlineReply service parameters	68
Table 28 – EM_GetDeviceAttribute service parameters.....	69
Table 29 – EM_ActiveNotification service parameters.....	71
Table 30 – EM_ConfiguringDevice service primitives.....	72
Table 31 – EM_SetDefaultValue service parameter	74
Table 32 – Parameters for domain download service.....	76
Table 33 – Parameters for domain upload service.....	78
Table 34 – EventReport service parameters.....	79
Table 35 – AcknowledgeEventReport service parameters	80
Table 36 – ReportConditionChanging service parameters.....	81
Table 37 – Read service parameters.....	82
Table 38 – Write service parameters.....	83
Table 39 – VariableDistribute service parameters	84
Table 40 – FRTVariableDistribute service parameters.....	85
Table 41 – FRTRead service parameters	86
Table 42 – FRTWrite service parameters	87
Table 43 – Summary of application layer services.....	91

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-14: Application layer service definition – Type 14 elements

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

NOTE 1 Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in type combinations as specified explicitly in the profile parts. Use of the various protocol types in other combinations may require permission of their respective intellectual-property-right holders.

International Standard IEC 61158-5-14 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2007. This edition constitutes a technical revision.

The main changes with respect to the previous edition are listed below:

- updated Normative references and Bibliography;

- corrections of the edit error;
- specification changes for CPF3;
- update of the requirements for all conformance classes;
- update of the requirements for all conformance services.

This bilingual version published in 2012-01 corresponds to the English version published in 2010-08.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/606/FDIS	65C/620/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The French version has not been voted upon.

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

A list of all parts of the IEC 61158 series, published under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

NOTE 2 The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC/TR 61158-1.

The application service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This standard defines the application service characteristics that fieldbus applications and/or system management may exploit.

Throughout the set of fieldbus standards, the term “service” refers to the abstract capability provided by one layer of the OSI Basic Reference Model to the layer immediately above. Thus, the application layer service defined in this standard is a conceptual architectural service, independent of administrative and implementation divisions.

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2010

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-14: Application layer service definition – Type 14 elements

1 Scope

1.1 Overview

The fieldbus application layer (FAL) provides user programs with a means to access the fieldbus communication environment. In this respect, the FAL can be viewed as a “window between corresponding application programs.”

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 14 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to 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 standard defines in an abstract way the externally visible service provided by the Type 14 fieldbus application layer in terms of

- a) an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,
- b) the primitive actions and events of the service;
- c) the parameters associated with each primitive action and event, and the form which they take; and
- d) the interrelationship between these actions and events, and their valid sequences.

The purpose of this standard is to define the services provided to

- a) the FAL user at the boundary between the user and the application layer of the fieldbus reference model, and
- b) Systems Management at the boundary between the application layer and Systems Management of the fieldbus reference model.

This standard specifies the structure and services of the Type 14 fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI application layer structure (ISO/IEC 9545).

FAL services and protocols are provided by FAL application-entities (AE) contained within the application processes. The FAL AE is composed of a set of object-oriented application service elements (ASEs) and a layer management entity (LME) that manages the AE. The ASEs provide communication services that operate on a set of related application process object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can

send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this standard is to specify the characteristics of conceptual application layer services suitable for time-critical communications, and thus supplement the OSI Basic Reference Model in guiding the development of application layer protocols for time-critical communications.

A secondary objective is to provide migration paths from previously-existing industrial communications protocols. It is this latter objective which gives rise to the diversity of services standardized as the various types of IEC 61158.

This specification may be used as the basis for formal Application Programming Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this specification, including

- a) the sizes and octet ordering of various multi-octet service parameters, and
- b) the correlation of paired request and confirm, or indication and response, primitives.

1.3 Conformance

This standard does not specify individual implementations or products, nor do they constrain the implementations of application layer entities within industrial automation systems.

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfill the Type 14 application layer services as defined in this standard.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-4-14, *Industrial communication networks – Fieldbus specifications – Part 4-14: Data-link layer protocol specification – Type 14 elements*

IEC 61588, *Precision clock synchronization protocol for networked measurement and control systems*

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 8822, *Information technology – Open Systems Interconnection – Presentation service definition*

ISO/IEC 8824-1, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ISO/IEC 9545, *Information technology – Open Systems Interconnection – Application Layer structure*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

3 Terms, definitions, symbols, abbreviations and conventions

For the purposes of this document, the following terms as defined in these publications apply:

3.1 ISO/IEC 7498-1 terms

- 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

3.2 ISO/IEC 8822 terms

- a) abstract syntax
- b) presentation context

3.3 ISO/IEC 9545 terms

- 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

3.4 ISO/IEC 8824-1 terms

- a) object identifier
- b) type

3.5 Fieldbus application-layer specific definitions

3.5.1

access control

control on the reading and writing of an object

3.5.2

access Path

association of a symbolic name with a variable for the purpose of open communication

3.5.3

communication macrocycle

set of basic cycles needed for a configured communication activity in a macro network segment

3.5.4

communication scheduling

algorithms and operation for data transfers occurring in a deterministic and repeatable manner

3.5.5

configuration (of a system or device)

step in system design: selecting functional units, assigning their locations and defining their interconnections

3.5.6

cyclic

repetitive in a regular manner

3.5.7

destination FB Instance

FB instance that receives the specified parameters

3.5.8

domain

part of memory used to store code or data

3.5.9

domain download

operation to write data in a domain

3.5.10

domain upload

operation to read data from a domain

3.5.11

entity

particular thing, such as a person, place, process, object, concept, association, or event

3.5.12

bridge

DL-relay entity which performs synchronization between links (buses) and may perform selective store-and-forward and routing functions to connect two micro network segments

3.5.13

identifier

16-bit word associated with a system variable

3.5.14

index

address of an object within an application process

3.5.15

instance

actual physical occurrence of an object within a class that identifies one of many objects within the same object class

3.5.16**instantiation**

creation of an instance of a specified type

3.5.17**management information**

network-visible information for the purpose of managing the field system

3.5.18**management information base**

organized list of management information

3.5.19**mapping**

set of values having defined correspondence with the quantities or values of another set

3.5.20**message filtering**

decision on a message according to a special rule

3.5.21**micro segment**

part of a network, where special scheduling is implemented

3.5.22**offset**

number of octets from a specially designated position

3.5.23**phase**

elapsed fraction of a cycle, measured from some fixed origin

3.5.24**process interface**

data exchange and information mapping between physical process and application unit

3.5.25**real-time**

ability of a system to provide a required result in a bounded time

3.5.26**real-time communication**

transfer of data in real-time

3.5.27**real-time Ethernet (RTE)**

ISO/IEC 8802-3-based network that includes real-time communication

NOTE 1 Other communication can be supported, providing the real-time communication is not compromised.

NOTE 2 This definition is dedicated, but not limited, to ISO/IEC 8802-3. It could be applicable to other IEEE 802 specifications, for example IEEE 802.11.

3.5.28**schedule**

temporal arrangement of a number of related operations

3.5.29

scheduling macrocycle

time interval to implement a specific schedule

3.5.30

source FB Instance

FB instance that sends a specific parameter

3.5.31

time offset

time difference from a specially designated time

3.6 Abbreviations and symbols

AAE	Application Access Entity
AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
APO	Application Object
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
AR	Application Relationship
ARP	Address Resolution Protocol
AREP	Application Relationship End Point
ASE	Application Service Element
Cnf	Confirmation
CR	Communication Relationship
CREP	Communication Relationship End Point
CSMA/CD	Carrier Sense Multiple Access Protocol with Collision Detection
DD	Device Description
DHCP	Dynamic Host Configuration Protocol
DL-	(as a prefix) Data Link-
DLCEP	Data Link Connection End Point
DLL	Data Link Layer
DLE	Data Link Entity
DLM	Data Link-management
DLS	Data Link Service
DLSAP	Data Link Service Access Point
DLSDU	DL-service-data-unit
ECSME	Type 14 communication scheduling management entity
Type 14	Ethernet for Plant Automation
EM_	(as a prefix) Type 14 Management
ESME	Type 14 Socket Mapping Entity
FB	Function Block
FBAP	Function Block Application Process
FRT	Fast Real-time
Ind	Indication

IP	Internet Protocol
LLC	Logical Link Control
LMP	Link Management Protocol
MAC	Medium Access Control
MAU	Medium Attachment Unit
MOB	Management Object Base
PAD	Pad (bits)
PDU	Protocol Data Unit
P/S	Publisher/Subscriber
Req	Request
Rsp	Response
RTE	Real-Time Ethernet
RT-Ethernet	Real-Time Ethernet
SAP	Service Access Point
SDU	Service Data Unit
SME	System Management Entity
SNTP	Simple Network Time Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

3.7 Conventions

3.7.1 Overview

The FAL is defined as a set of object-oriented ASEs. Each ASE is specified in a separate subclause. Each ASE specification is composed of two parts, its class specification, and its service specification.

The class specification defines the attributes of the class. The attributes are accessible from instances of the class using the Object Management ASE services specified in Clause 5. The service specification defines the services that are provided by the ASE.

3.7.2 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is shown below:

FAL ASE:	ASE Name
CLASS:	Class name
CLASS ID:	#
PARENT CLASS:	Parent class name
ATTRIBUTES:	
1	(o) Key Attribute: numeric identifier
2	(o) Key Attribute: name
3	(m) Attribute: attribute name(values)
4	(m) Attribute: attribute name(values)
4.1	(s) Attribute: attribute name(values)
4.2	(s) Attribute: attribute name(values)
4.3	(s) Attribute: attribute name(values)
5.	(c) Constraint: constraint expression
5.1	(m) Attribute: attribute name(values)
5.2	(o) Attribute: attribute name(values)
6	(m) Attribute: attribute name(values)

6.1 (s) Attribute: attribute name(values)

6.2 (s) Attribute: attribute name(values)

SERVICES:

1 (o) OpsService: service name

2. (c) Constraint: constraint expression

2.1 (o) OpsService: service name

3 (m) MgtService: service name

(a) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.

(b) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.

(c) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2048 are allocated for identifying user defined classes.

(c) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

NOTE The parent-class "TOP" indicates that the class being defined is an initial class definition. The parent class TOP is used as a starting point from which all other classes are defined. The use of TOP is reserved for classes defined by this standard.

(d) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.

1) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.

2) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.

3) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify

i) fields of a structured attribute (4.1, 4.2, 4.3),

ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).

iii) the selection fields of a choice type attribute (6.1 and 6.2).

(e) The "SERVICES" label indicates that the following entries are services defined for the class.

1) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.

2) The label "OpsService" designates an operational service (1).

3) The label "MgtService" designates an management service (2).

- 4) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.7.3 Conventions for service definitions

3.7.3.1 General

This standard uses the descriptive conventions given in ISO/IEC 10731.

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

3.7.3.2 Service parameters

Service primitives are used to represent service user/service provider interactions (ISO/IEC 10731). They convey parameters which indicate information available in the user/provider interaction.

NOTE 1 See the note under 3.7.3.3 relative to the non-inclusion of service parameters that are appropriate to a protocol specification or programming interface specification or implementation specification, but not to an abstract service definition.

This standard uses a tabular format to describe the component parameters of the service primitives. The parameters that apply to each group of service primitives are set out in tables throughout the remainder of this standard. Each table consists of up to six columns: a column for the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the service. The possible six columns are:

- a) the parameter name;
- b) the request primitive's input parameters;
- c) the request primitive's output parameters;

NOTE 2 This is a seldom-used capability. Unless otherwise specified, request primitive parameters are input parameters.

- d) the indication primitive's output parameters;
- e) the response primitive's input parameters; and
- f) the confirm primitive's output parameters.

NOTE 3 The request, indication, response and confirm primitives are also known as requestor.submit, acceptor.deliver, acceptor.submit, and requestor.deliver primitives, respectively (see ISO/IEC 10731).

The service specifications of this standard uses a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

- a) Parameter name,
- b) request primitive,
- c) indication primitive,
- d) response primitive, and
- e) confirm primitive.

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column:

- M parameter is mandatory for the primitive
- U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.

- C parameter is conditional upon other parameters or upon the environment of the service user.
- (blank) parameter is never present.
- S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

- a parameter-specific constraint:
“(=)” indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.
- an indication that some note applies to the entry:
“(n)” indicates that the following note "n" contains additional information pertaining to the parameter and its use.

3.7.3.3 Service procedures

The procedures are defined in terms of

- the interactions between application entities through the exchange of fieldbus Application Protocol Data Units, and
- the interactions between an application layer service provider and an application layer service user in the same system through the invocation of application layer service primitives.

These procedures are applicable to instances of communication between systems which support time-constrained communications services within the fieldbus application layer.

NOTE The IEC 61158-5 subseries of standards define sets of abstract services. They are neither protocol specifications nor implementation specifications nor concrete programming interface specifications. Therefore there are restrictions on the extent to which service procedures can be mandated in the parts of IEC 61158-5 subseries. Protocol aspects that can vary among different protocol specifications or different implementations that instantiate the same abstract services are unsuitable for inclusion in these service definitions, except at the level of abstraction that is necessarily common to all such expressions.

For example, the means by which service providers pair request and reply PDUs is appropriate for specification in an IEC 61158-6 protocol specification standard but not in an IEC 61158-5 subseries abstract service definition standard. Similarly, local implementation methods by which a service provider or service user pairs request and confirm primitives, or indication and response primitives, is appropriate for an implementation specification or for a programming interface specification, but not for an abstract service standard or for a protocol standard, except at a level of abstraction that is necessarily common to all embodiments of the specifying standard. In all cases, the abstract definition is not permitted to over-specify the more concrete instantiating realization.

Further information on the conceptual service procedures of an implementation of a protocol that realizes the services of one of the IEC 61158-5 subseries abstract service definitions can be found in IEC/TR 61158-1:2007, 9.6.

4 Concepts

The common concepts and templates used to describe the application layer service in this standard are detailed in IEC/TR 61158-1, Clause 9.

5 Data type ASE

5.1 Overview

An overview of the data type ASE and the relationships between data types is provided in IEC/TR 61158-1, 10.1.

5.2 Formal definition of data type objects

The template used to describe the data type class in this clause is detailed in IEC/TR 61158-1, 10.2. This includes the specific ASE structure and the definition of its attributes.

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

5.3.1.1.1 Boolean

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 1
2	Data type Name = Boolean
3	Format = FIXED LENGTH
4.1	Octet Length = 1

This data type expresses a Boolean data type with the values TRUE and FALSE.

5.3.1.1.2 BOOL

This IEC 61131-3 type is the same as Boolean.

5.3.1.1.3 VT_BOOLEAN

CLASS:	Data type
ATTRIBUTES:	
2	Data type Name = VT_BOOLEAN
4	Format = FIXED LENGTH
4.1	Octet Length = 2

This data type expresses a Boolean data type with the values TRUE (-1) and FALSE (0) (see Interger16).

5.3.1.2 Bitstring types

5.3.1.2.1 BitString8

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 22
2	Data type Name = Bitstring8
3	Format = FIXED LENGTH
5.1	Octet Length = 1

This type contains 1 element of type BitString.

5.3.1.2.2 OCTET

This IEC 61131-3 type is the same as Bitstring8.

5.3.1.2.3 BitString16

CLASS:	Data type
ATTRIBUTES:	
1	Data type Numeric Identifier = 23
2	Data type Name = Bitstring16
3	Format = FIXED LENGTH
5.1	Octet Length = 2

5.3.1.2.4 WORD

This IEC 61131-3 type is the same as Bitstring16.

5.3.1.2.5 BitString32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	24
2	Data type Name	=	Bitstring32
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	4

5.3.1.2.6 DWORD

This IEC 61131-3 type is the same as Bitstring32.

5.3.1.2.7 BitString64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	57
2	Data type Name	=	Bitstring64
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	8

5.3.1.2.8 LWORD

This IEC 61131-3 type is the same as Bitstring64.

5.3.1.3 Date types

5.3.1.3.1 BinaryDate

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	11
2	Data type Name	=	BinaryDate
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

This data type is composed of six elements of unsigned values and expresses calendar date and time. The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds. The second element is an Unsigned8 data type and gives the fraction of an hour in minutes. The third element is an Unsigned8 data type and gives the fraction of a day in hours. The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of the week and its lower five (5) bits give the day of the month. The fifth element is an Unsigned8 data type and gives the month. The last element is Unsigned8 data type and gives the year.

5.3.1.3.2 BinaryDate2000

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	51
2	Data type Name	=	BinaryDate2000
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This data type is composed of six elements of unsigned values and expresses calendar date and time. The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds. The second element is an Unsigned8 data type and gives the fraction of an hour in minutes. The third element is an Unsigned8 data type and gives the fraction of a day in hours. The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of

the week and its lower five (5) bits give the day of the month. The fifth element is an Unsigned8 data type and gives the month. The last element is Unsigned16 data type and gives the year.

5.3.1.3.3 Date

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	50
2	Data type Name	=	Date
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

This data type is composed of six elements of unsigned values and expresses calendar date and time. The first element is an Unsigned16 data type and gives the fraction of a minute in milliseconds. The second element is an Unsigned8 data type and gives the fraction of an hour in minutes. The third element is an Unsigned8 data type and gives the fraction of a day in hours with the most significant bit indicating Standard Time or Daylight Saving Time. The fourth element is an Unsigned8 data type. Its upper three (3) bits give the day of the week and its lower five (5) bits give the day of the month. The fifth element is an Unsigned8 data type and gives the month. The last element is Unsigned8 data type and gives the year. The values 0 ... 50 correspond to the years 2000 to 2050, the values 51 ... 99 correspond to the years 1951 to 1999.

5.3.1.3.4 DATE

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	DATE
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This IEC 61131-3 type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets. It expresses the date as a number of days, starting from 1972.01.01 (January 1st, 1972), the start of the Coordinated Universal Time (UTC) era, until 2151.06.06 (June 6th, 2151), i.e. a total range of 65536 days.

5.3.1.3.5 date

This data type is the same as Float64.

The data type date has a resolution in the range of one nanosecond. It is valid for dates between 1 January 0100 and 31 December 9999. The value 0,0 has been defined for 30 December 1899, 00:00. The integer part of the value represents the days after 30 December 1899 (for dates before this day, the corresponding value is negative); the fractional part defines the time at that day.

5.3.1.3.6 TimeOfDay

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	12
2	Data type Name	=	TimeOfDay
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

This data type is composed of two elements of unsigned values and expresses the time of day and the date. The first element is an Unsigned32 data type and gives the time after the

midnight in milliseconds. The second element is an Unsigned16 data type and gives the date counting the days from January 1, 1984.

5.3.1.3.7 TimeOfDay with date indication

This data type is the same as the TimeOfDay data type defined above.

5.3.1.3.8 TimeOfDay without date indication

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|-----------------------------------|
| 1 | Data type Numeric Identifier | = | 52 |
| 2 | Data type Name | = | TimeOfDay without date indication |
| 4 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

This data type is composed of one element of an unsigned value and expresses the time of day. The element is an Unsigned32 data type and gives the time after the midnight in milliseconds.

5.3.1.3.9 TIME_OF_DAY

This IEC 61131-3 type is the same as TimeOfDay without date indication.

5.3.1.3.10 PrecisionTimeDifference

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|-------------------------|
| 1 | Data type Numeric Identifier | = | 13 |
| 2 | Data type Name | = | PrecisionTimeDifference |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 or 8 |

This data type is composed of two elements of unsigned values that expresses the difference in time. The first optional element is an Unsigned32 data type that provides the difference in seconds. The last element is an signed32 data type that provides the fractional portion of one second in microseconds.

5.3.1.3.11 PrecisionTimeDifference with second indication

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--|
| 1 | Data type Numeric Identifier | = | 53 |
| 2 | Data type Name | = | PrecisionTimeDifference with second indication |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 8 |

This data type is composed of two elements of unsigned values that expresses the difference in time. The first element is an Unsigned32 data type that provides the difference in seconds. The last element is an signed32 data type that provides the fractional portion of one second in microseconds.

5.3.1.3.12 PrecisionTimeDifference without second indication

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|---|
| 1 | Data type Numeric Identifier | = | 54 |
| 2 | Data type Name | = | PrecisionTimeDifference without second indication |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

This data type is a signed value that expresses the difference in time in microseconds.

5.3.1.3.13 TimeValue

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	21
2	Data type Name	=	Time Value
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This simple type expresses the time or time difference in a two's complement binary number with a length of eight octets. The unit of time is 1/32 millisecond.

5.3.1.3.14 UniversalTime

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	16
2	Data type Name	=	UniversalTime
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	12

This simple type is composed of twelve elements of type VisibleString. (YYMMDDHHMMSS). It is the same as that defined in ISO/IEC 8824-1, except that the local time differential is not supported.

5.3.1.3.15 FieldbusTime

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	17
2	Data type Name	=	FieldbusTime
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

This data type is defined in IEC 61158-4-14 of this standard as DL-Time.

5.3.1.4 Numeric types

5.3.1.4.1 BCD

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	35
2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of one octet. In this type, the least significant four bits are used to express a BCD value which is between zero and nine inclusive. The most significant four bits are unused.

5.3.1.4.2 Floating Point types

5.3.1.4.2.1 Float32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	8
---	------------------------------	---	---

2	Data type Name	=	Float32
4	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type has a length of four octets. The format for float32 is that defined by ANSI/IEEE 754 as single precision.

5.3.1.4.2.2 Floating point

This data type is the same as Float32.

5.3.1.4.2.3 REAL

This IEC 61131-3 type is the same as Float32.

5.3.1.4.2.4 float

This data type is the same as Float32.

5.3.1.4.2.5 Float64

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	15
2	Data type Name	=	Float64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This type has a length of eight octets. The format for float64 is that defined by ANSI/IEEE 754 as double precision.

5.3.1.4.2.6 LREAL

This IEC 61131-3 type is the same as Float64.

5.3.1.4.2.7 double

This data type is the same as Float64.

5.3.1.4.3 Integer types

5.3.1.4.3.1 Integer8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	2
2	Data type Name	=	Integer8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This integer type is a two's complement binary number with a length of one octet.

5.3.1.4.3.2 SINT

This IEC 61131-3 type is the same as Integer8.

5.3.1.4.3.3 char

This data type is the same as Integer8.

5.3.1.4.3.4 Integer16**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	3
2	Data type Name	=	Integer16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This integer type is a two's complement binary number with a length of two octets.

5.3.1.4.3.5 INT

This IEC 61131-3 type is the same as Integer16.

5.3.1.4.3.6 short

This data type is the same as Integer16.

5.3.1.4.3.7 Integer32**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	4
2	Data type Name	=	Integer32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This integer type is a two's complement binary number with a length of four octets.

5.3.1.4.3.8 DINT

This IEC 61131-3 type is the same as Integer32.

5.3.1.4.3.9 long

This data type is the same as Integer32.

5.3.1.4.3.10 Integer64**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	55
2	Data type Name	=	Integer64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This integer type is a two's complement binary number with a length of eight octets.

5.3.1.4.3.11 LINT

This IEC 61131-3 type is the same as Integer64.

5.3.1.4.4 Unsigned types**5.3.1.4.4.1 Unsigned8****CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	5
---	------------------------------	---	---

2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of one octet.

5.3.1.4.4.2 USINT

This IEC 61131-3 type is the same as Unsigned8.

5.3.1.4.4.3 unsigned char

This data type is the same as Unsigned8.

5.3.1.4.4.4 Unsigned16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	6
2	Data type Name	=	Unsigned16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of two octets.

5.3.1.4.4.5 UINT

This IEC 61131-3 type is the same as Unsigned16.

5.3.1.4.4.6 unsigned short

This data type is the same as Unsigned16.

5.3.1.4.4.7 Unsigned32

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	7
2	Data type Name	=	Unsigned32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of four octets.

5.3.1.4.4.8 UDINT

This IEC 61131-3 type is the same as Unsigned32.

5.3.1.4.4.9 unsigned long

This data type is the same as Unsigned32.

5.3.1.4.4.10 Unsigned64**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 56
- 2 Data type Name = Unsigned64
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 8

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This unsigned type has a length of eight octets.

5.3.1.4.4.11 ULINT

This IEC 61131-3 type is the same as Unsigned64.

5.3.1.4.4.12 Normalised 2 octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 113
- 2 Data type Name = N2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Linear normalized value. 0% corresponds to 0 (0x0), 100% corresponds to 214 (0x4000).

Representation in two's complement, the MSB (Most Significant Bit) is the bit after the sign bit (SN) of the first octet. The attribute of data type and the weight of each bit are shown in Table 1 and Table 2.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

Table 1 – Attribute of variable normalised 2 octet

Data type	Range of values	Resolution	Length
N2	$-200 \% \leq i \leq (200 \cdot 2^{-14}) \%$	$2^{-14} = 0.0061 \%$	2 Octets

Table 2 – Encoding of variable normalised 2 octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}

5.3.1.4.4.13 Normalised 4 octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 114
- 2 Data type Name = N4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Linear normalized value. 0% corresponds to 0 (0x0), 100% corresponds to 230 (0x4000 0000).

Representation in two's complement, the MSB (Most Significant Bit) is the bit after the sign bit (SN) of the first octet. The attribute of data type and the weight of each bit are shown in Table 3 and Table 4.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

Table 3 – Attribute of normalised 4 Octet

Data type	Range of values	Resolution	Length
N4	$-200 \% \leq i \leq (200-2^{-30}) \%$	$2^{-30} = 9.3 * 10^{-8} \%$	4 Octet

Table 4 – Encoding of normalised 4 Octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2				:				
3				:				
4	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}

5.3.1.4.4.14 Variable normalised 2 octet

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 123
- 2 Data type Name = X2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Linear normalized value. 0% corresponds to 0 (0x0), 100% corresponds to 2X. The structure is the same as for data types N2 and N4, but the normalization (100 %) does not automatically refer to bit 14 or bit 30 respectively, but is variable. The normalization bit is coded in an additional parameter.

Representation in two's complement, the MSB (Most Significant Bit) is the bit after the sign bit (SN) of the first octet. The attribute of data type and the weight of each bit are shown in Table 5 and Table 6.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

Table 5 – Attribute of variable normalised 2 octet

Data type	Range of values	Resolution	Length
X2 (e.g. with x = 12)	$-800 \% \leq i \leq (800-2^{-14}) \%$	2^{-12}	2 Octet

Table 6 – Encoding of variable normalised 2 octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}

5.3.1.4.4.15 Variable Normalised 4 Octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 124
 2 Data type Name = X4
 3 Format = FIXED LENGTH
 4.1 Octet Length = 4

Linear normalized value. 0% corresponds to 0 (0x0), 100% corresponds to 2X. The structure is the same as for data types N2 and N4, but the normalization (100 %) does not automatically refer to bit 14 or bit 30 respectively, but is variable. The normalization bit is coded in an additional parameter.

Representation in two's complement, the MSB (Most Significant Bit) is the bit after the sign bit (SN) of the first octet. The attribute of data type and the weight of each bit are shown in Table 7 and Table 8.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

Table 7 – Attribute of variable normalised 4 Octet

Data type	Range of values	Resolution	Length
X4 (e.g. with x = 28)	$-800 \% \leq i \leq (800 \cdot 2^{-30}) \%$	2^{-28}	2 Octet

Table 8 – Encoding of variable normalised 4 Octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2				:				
3				:				
4	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}

5.3.1.4.4.16 Unipolar 2 Octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 125
 2 Data type Name = Unipolar2.16
 3 Format = FIXED LENGTH
 4.1 Octet Length = 2

A primitive type with distinguished values which are non-negative, whole numbers divided by a fixed power of two, expressing a value in percent of a span.

NOTE 1 These types do not exist in ASN.1, they are expressed in IEC 870 as 'unsigned fixed point number'

NOTE 2 These types is defined in IEC 61375

The number before the comma gives the number of power of 2 forming the integer part. The epsilon factor is equal to the value of the smallest power of two in the word. A variable of unipolar type should be transmitted as an unsigned integer. The attribute of data type and the weight of each bit are shown in Table 9 and Table 10.

Table 9 – Attribute of unipolar 2 octet

Data type	Range of values	Resolution	Length
Unipolar2.16	$0 \% \leq i \leq (400 \cdot 2^{-14}) \%$	2^{-14}	2 Octet

Table 10 – Encoding of unipolar 2 octet

Octet	Bit															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1,2	2 ¹	2 ⁰	2 ⁻¹	2 ⁻²	2 ⁻³	2 ⁻⁴	2 ⁻⁵	2 ⁻⁶	2 ⁻⁷	2 ⁻⁸	2 ⁻⁹	2 ⁻¹⁰	2 ⁻¹¹	2 ⁻¹²	2 ⁻¹³	2 ⁻¹⁴
Integer part	Fractional part															

5.3.1.4.4.17 Fixed point value 2 Octet

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 121
- 2 Data type Name = E2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Linear fixed-point value with seven binary places after the decimal point. 0 corresponds to 0 (0x0), 128 corresponds to 214 (0x4000).

Representation in two's complement, the MSB (Most Significant Bit) is the bit after the sign bit (SN) of the first octet. The attribute of data type and the weight of each bit are shown in Table 11 and Table 12.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

Table 11 – Attribute of Fixed point value 2 Octet

Data type	Range of values	Resolution	Length
E2	$-256 \cdot 2^{-7} \leq i \leq 256 \cdot 2^{-7}$	$2^{-7} = 0,007\ 812\ 5$	2 Octet

Table 12 – Encoding of Fixed point value 2 Octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
2	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

5.3.1.4.4.18 Fixed point value 4 Octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 122
 2 Data type Name = C4
 3 Format = FIXED LENGTH
 4.1 Octet Length = 4

Linear fixed point value with four decimal places. 0 corresponds to 0 (0x0), 0.0001 corresponds to 2^0 (0x0000 0001).

As by Integer32, weighting of the bits has been reduced by a factor of 10000. The attribute of data type is shown in Table 13.

Table 13 – Attribute of Fixed point value 4 Octet

Data type	Range of values	Resolution	Length
C4	$-214\,748,364\,8 \leq i \leq 214\,748,364\,7$	$10^{-4} = 0.0001$	4 Octet

5.3.1.4.4.19 Bit sequence 2 Octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 115
 2 Data type Name = V2
 3 Format = FIXED LENGTH
 4.1 Octet Length = 2

Bit sequence to control and represent application functions. 16 Boolean variables are combined in two octets. The weight of each bit is shown in Table 14.

Table 14 – Encoding of Bit sequence 2 Octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	15	14	13	12	11	10	9	8
2	7	6	5	4	3	2	1	0

5.3.1.4.4.20 Nibble 4 Octet**CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 116
 2 Data type Name = L2
 3 Format = FIXED LENGTH

4.1 Octet Length = 4

Four associated bits form a nibble. Four nibbles are represented in two octets.

The definition of the nibble is not specified. The weight of each bit is shown in Table 15.

Table 15 – Encoding of Nibble 4 Octet

Octet	Bit							
	7	6	5	4	3	2	1	0
1	Nibble 3				Nibble 2			
2	Nibble 1				Nibble 0			

5.3.1.5 OctetString character types

5.3.1.5.1 OctetString1

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 30
- 2 Data type Name = OctetString1
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 1

This type has a length of one octet.

5.3.1.5.2 OctetString2

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 31
- 2 Data type Name = OctetString2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

This type has a length of two octets.

5.3.1.5.3 OctetString4

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 32
- 2 Data type Name = OctetString4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

This type has a length of four octets.

5.3.1.5.4 OctetString8

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 33
- 2 Data type Name = OctetString8
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 8

This octet string type has a length of eight octets.

5.3.1.5.5 OctetString16**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	34
2	Data type Name	=	OctetString16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	16

This type has a length of 16 octets.

5.3.1.6 Pointer types**5.3.1.6.1 Interface pointer****CLASS:** Data type**ATTRIBUTES:**

2	Data type Name =	Interface Pointer
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This data type defines a 4 octet fixed length data type.

5.3.1.6.2 LPWSTR**CLASS:** Data type**ATTRIBUTES:**

2	Data type Name =	LPWSTR
3	Format	= FIXED LENGTH
4.1	Octet Length	= 4

This data type defines a reference to an UnicodeString.

5.3.1.7 Time types**5.3.1.7.1 BinaryTime0****CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	40
2	Data type Name	=	BinaryTime0
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of two octets. The unit of time for this type is 10 µs.

5.3.1.7.2 BinaryTime1**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	41
2	Data type Name	=	BinaryTime1
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of two octets. The unit of time for this type is 100 µs.

5.3.1.7.3 BinaryTime2

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	42
2	Data type Name	=	BinaryTime2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of two octets. The unit of time for this type is 1 ms.

5.3.1.7.4 BinaryTime3

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	43
2	Data type Name	=	BinaryTime3
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of two octets. The unit of time for this type is 10 ms.

5.3.1.7.5 BinaryTime4

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	44
2	Data type Name	=	BinaryTime4
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of four octets. The unit of time for this type is 10 µs.

5.3.1.7.6 BinaryTime5

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	45
2	Data type Name	=	BinaryTime5
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of four octets. The unit of time for this type is 100 µs.

5.3.1.7.7 BinaryTime6

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	46
2	Data type Name	=	BinaryTime6
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of four octets. The unit of time for this type is 1 ms.

5.3.1.7.8 BinaryTime7

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	47
2	Data type Name	=	BinaryTime7
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of six octets. The unit of time for this type is 1 ms.

5.3.1.7.9 BinaryTime8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	48
2	Data type Name	=	BinaryTime8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This type has a length of six octets. The unit of time for this type is 10 μ s.

5.3.1.7.10 BinaryTime9

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	49
2	Data type Name	=	BinaryTime9
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

This type is a binary number. The most significant bit of the most significant octet is always used as the most significant bit of the binary number; no sign bit is included. This binary time type has a length of six octets. The unit of time for this type is 100 μ s.

5.3.1.7.11 TIME

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	TIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This IEC 61131-3 type is a two's complement binary number with a length of four octets. The unit of time for this type is 1 ms.

5.3.1.7.12 ITIME

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
---	------------------------------	---	----------

2	Data type Name	=	ITIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This IEC 61131-3 type extension is a two's complement binary number with a length of two octets. The unit of time for this type is 1 ms.

5.3.1.7.13 FTIME

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	FTIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This IEC 61131-3 type extension is a two's complement binary number with a length of four octets. The unit of time for this type is 1 µs.

5.3.1.7.14 LTIME

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	not used
2	Data type Name	=	LTIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This IEC 61131-3 type extension is a two's complement binary number with a length of eight octets. The unit of time for this type is 1 µs.

5.3.1.7.15 NetworkTime

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	58
2	Data type Name	=	NetworkTime
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This data type is composed of two unsigned values that express the network time.

The first element is an Unsigned32 data type that provides the network time in seconds since 1900.01.01 00:00:00(UTC) for network time greater/equal 1984.01.01 00:00:00 (UTC) and less than 2036.07.02 06:28:16(UTC), or in seconds since 2036.07.02 06:28:16(UTC) for Network time greater or equal 2036.07.02 06:28:16(UTC).

The second element is an Unsigned32 data type that provides the fractional portion of seconds in $1/2^{32}$ s).

5.3.1.7.16 NetworkTimeDifference

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	59
2	Data type Name	=	NetworkTimeDifference
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This data type is composed of an integer value and of an unsigned value that express the difference in network time. The first element is an Integer32 data type that provides the

network time difference in seconds. The second element is an Unsigned32 data type that provides the fractional portion of seconds in $1/2^{32}$ s.

5.3.1.7.17 Multiple time constant 2 octets

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 118
- 2 Data type Name = T2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Time data as a multiple of constant sampling time T_a .

As by Unsigned16 with a restricted Range of values $0 \leq x \leq 32767$. When interpreted, internal values outside this range are set to 0. The attribute of data type is shown in Table 16.

Table 16 – Attribute of multiple time constant 2 octets

Data type	Range of values	Resolution	Length
T2	$0 \leq i \leq 32\,767 \times T_a$	T_a	2 Octet

NOTE The value of this time parameter refers to the additional specified constant sampling time T_a . The associated sampling time is required in order to interpret the internal value.

5.3.1.7.18 Multiple time constant 4 octets

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 119
- 2 Data type Name = T4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Time data as a multiple of constant sampling time T_a .

As by Unsigned32 with a restricted Range of values $0 \leq x \leq 4294967295$. When interpreted, internal values outside this range are set to 0. The attribute of data type is shown in Table 17.

Table 17 – Attribute of multiple time constant 4 octets

Data type	Range of values	Resolution	Length
T2	$0 \leq i \leq 4\,294\,967\,295 \times T_a$	T_a	4 Octet

NOTE The value of this time parameter refers to the additional specified constant sampling time T_a . The associated sampling time is required in order to interpret the internal value.

5.3.1.7.19 Fraction time constant 2 octets

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 120
- 2 Data type Name = D2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Time data as a fraction of the constant sampling time T_a .

As by Unsigned16 with a restricted Range of values $0 \leq x \leq 32767$. When interpreted, internal values outside this range are set to 0. Interpreted value = Internal value * $T_a/16384$. The attribute of data type is shown in Table 18.

Table 18 – Attribute of fraction time constant 2 octets

Data type	Range of values	Resolution	Length
D2	$0 \leq i \leq (2 \cdot 2^{-14}) \times T_a$	$2^{-14} \times T_a$	2 Octet

NOTE The value of this time parameter refers to the additional specified constant sampling time T_a . The associated sampling time is required in order to interpret the internal value.

5.3.1.7.20 Reciprocal time constant 2 octets

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 120
- 2 Data type Name = D2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Time data as a reciprocal multiple of a constant sampling time T_a .

As by Unsigned16 with a limited Range of values $1 \leq x \leq 16384$. When interpreted, internal values outside this range are set to 16384. Interpreted value = $16384 \times T_a / \text{internal value}$. The attribute of data type is shown in Table 19.

Table 19 – Encoding of reciprocal time constant 2 octets

Data type	Range of values	Resolution	Length
R2	$1 \times T_a \leq i \leq 16384 \times T_a$	T_a	2 Octet

NOTE The value of this time parameter refers to the additional specified constant sampling time T_a . The associated sampling time is required in order to interpret the internal value.

5.3.1.8 VisibleString character types

5.3.1.8.1 UNICODE char

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 36
- 2 Data type Name = UnicodeChar
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

This type is defined as a single character in the UNICODE string type.

5.3.1.8.2 VisibleString1

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = 25
- 2 Data type Name = VisibleString1
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 1

This type is defined as a single character in the ISO VisibleString type.

5.3.1.8.3 VisibleString2**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	26
2	Data type Name	=	VisibleString2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

This type contains two elements of type VisibleString.

5.3.1.8.4 VisibleString4**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	27
2	Data type Name	=	VisibleString4
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type contains four elements of type VisibleString.

5.3.1.8.5 VisibleString8**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	28
2	Data type Name	=	VisibleString8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

This type contains eight elements of type VisibleString.

5.3.1.8.6 VisibleString16**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	29
2	Data type Name	=	VisibleString16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	16

This type contains 16 elements of type VisibleString.

5.3.2 String types**5.3.2.1 BitString****CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	14
2	Data type Name	=	Bitstring
3	Format	=	STRING
5.1	Octet Length	=	1 to n

This string type is defined as a series of BitString8 elements.

5.3.2.2 CompactBooleanArray**CLASS:** Data type**ATTRIBUTES:**

1	Data type Numeric Identifier	=	37
2	Data type Name	=	CompactBooleanArray
3	Format	=	STRING
6.1	Octet Length	=	1

In this type, each bit value of 0 (zero) represents the Boolean value FALSE and each bit value of 1 represents the Boolean value TRUE.

5.3.2.3 CompactBCDArray

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	38
2	Data type Name	=	CompactBCDArray
3	Format	=	STRING
4.1	Octet Length	=	1

This type is used to pack an ordered series of BCD values, two per octet, into an ordered series of octets. The first octet contains the most significant BCD value in its most significant quartet. If the number of BCD values is odd, the least significant quartet of the final octet is set to the reserved value "1111".

5.3.2.4 OctetString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	10
2	Data type Name	=	OctetString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

An OctetString is an ordered sequence of octets, numbered from 1 to n. For the purposes of discussion, octet 1 of the sequence is referred to as the first octet. IEC 61158-6 defines the order of transmission.

5.3.2.5 UNICODEString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	39
2	Data type Name	=	UnicodeString
3	Format	=	STRING
4.1	Octet Length	=	2

This type is defined as the UNICODE string type.

5.3.2.6 VisibleString

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	9
2	Data type Name	=	VisibleString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

This type is defined as the ISO/IEC 646 string type.

5.4 Data type ASE service specification

There are no operational services defined for the type object.

6 Communication model specification

6.1 General

The FAL AE is composed of FAL Management Entity (FME), Application Access Entity (AAE) and Socket Mapping Entity (SME), as shown in Figure 1.

FAL Management Entity and Services can support various management operations, including FAL Management ASE, AR ASE, etc.

AAE provides an interface for data communication between user application process, which is composed of domain ASE, variable ASE and event ASE.

SME provides an interface for AAE, FME and UDP/IP, which is composed of Socket Mapping ASE.

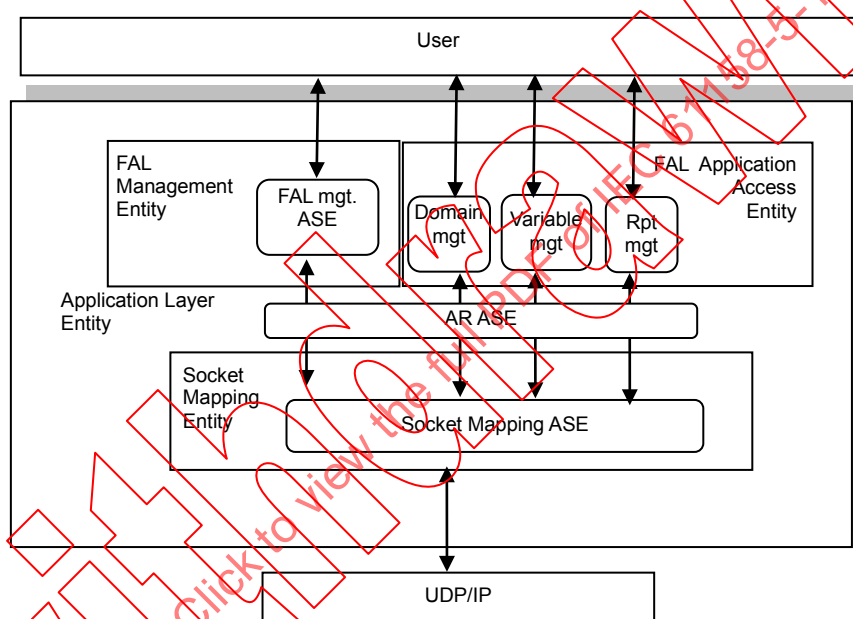


Figure 1 – Application layer entity

For FRT applications, AAE, FME can also access ECSME interface directly when meeting the follow conditions:

- the AL services called by AAE or FME shall be an unconfirmed service;
- communication between endpoints shall be restricted in one micro-segment.

6.2 ASEs

6.2.1 FAL management ASE

6.2.1.1 Overview

FAL Management is used to integrate several devices in the network into a harmonized communication system. FAL Management supports the function of device identification, address assignment, object location, clock synchronization and link management.

6.2.1.1.1 Device identification

A device can be identified by its physical device tag, device ID, device redundancy number (mutually redundancy devices).

Device ID

Device ID identifies a unique device. A unique Device ID is assigned to each individual device by the manufacturer. It is visible in FAL Management entity, but cannot be modified.

PD_Tag

For the purposes of device configuration, each field device has a physical device tag (PD_Tag). In a Type 14 system, PD_Tag is unique to identifier a physical device. Mutually redundancy device has the same PD_Tag.

Redundancy number

Mutual redundancy devices distinguish each other through the different redundancy number.

6.2.1.1.2 Address assignment

The IP address of each device can be statically appointed or dynamically assigned through Dynamic Host Configuration Protocol (DHCP). Once the IP address is assigned, it should be written into the non-volatile memory as permanent information until the device is reconfigured.

When the IP address confliction with another device is detected, the state of the local device should be changed to *No Address*, waiting to reset its IP address.

6.2.1.1.3 Object location

A Type 14 physical device can be located by IP address or PD_Tag. An FB instance in a Type 14 device can be located by FB_Tag or FB identifier AppID, which are unique in each device, but not unique in the whole network. Each parameter object of an FB instance can be located by parameter index, ObjectID, which is unique in each FB instance, but not unique in the whole device.

FAL Management entity provides EM_DetectingDevice service to query the IP address of a device using PD_Tag. It also provides EM_GetDeviceAttribute service to get the PD_Tag or other relevant information of a physical device.

EM_DetectingDevice service request primitive can be VariableDistributed in a network with unicast or multicast mode. The physical device receiving this request primitive will check its local PD_Tag. If local PD_Tag matches that in EM_DetectingDevice service request primitive, it will report its IP address, Device ID using EM_OnlineReply service.

EM_GetDeviceAttribute service request primitive can be delivered to a network using unicast mode.

6.2.1.1.4 Adding or deleting a device

After the power-on and initialization, if a device has no IP address, it enters *No Address* state to wait for IP address assignment. After IP address assignment, the device changes its state into *Unconfigured* and broadcasts a device annunciation message using EM_ActiveNotification service.

Once the user application receives the device annunciation message, it can configure the device attributes using EM_ConfiguringDevice service. After configuration, the device can operate in *Configured* state. The configuration information is written in the non-volatile memory as permanent information until device is reconfigured.

When powered-on again, the device can recover all configuration information and operate in *Configured* state automatically.

6.2.1.1.5 Clock synchronization

The time of each device should be synchronized. Either SNTP protocol defined in RFC2030 or IEC 61588 can be used according to practical application.

At each time of synchronization, local device sends a clock synchronization request primitive to time server at a configured time interval and keep synchronization with system time according to the response from time server.

The details of time synchronization are not specified in this specification (see RFC 2030 and IEC 61588).

6.2.1.1.6 Link management

The access path between FBs is defined as a link object. The link object specifies the communication relationship between the input/output of FBs. Link objects are configured using the configuration program.

Through a configured link object, an FB in a device can determine where to release its output data or where to receive updating input data.

6.2.1.1.7 Management object base

All the management objects using in FAL Management entity are organized in Management Object Base (MOB). The MOB is a two-dimension table as shown in Table 20. Each object has a unique identifier ObjectID.

As a special function block, the MOB has its default value of 0 for application identifier AppID for the purpose of reading/writing the objects. That is, other function block instances have a value greater than 0 for their application identifier AppID.

Table 20 – Management object base

Object	Object ID	Illustration
MIB Header	1	device Management Object Base header object
Device Descriptor	2	device descriptor object
Time Synchronization	3	Time Synchronization Object
Max Response Time	4	Confirmed Service Max Response Time
Communication Schedule Management	5	Communication Schedule Management Object
Device Application information	6	Device Application information Object
FB Application information Header	7	Function Block Application information Header
Link Object Header	8	Link Object Header
Domain Application Object Header	9	Domain Application Object Header
FRT Object Header	10	FRT Link Object Header
.....		
FB Application information 1	2000	Function Block Application information 1
FB Application information 2	2001	Function Block Application information 2
.....	Increased number in turn	
Domain Application Object 1	4000	Domain Application Object 1
Domain Application Object 2	4001	Domain Application Object 2
.....	Increased number in turn	
Link Object 1	5000	Link Object 1
Link Object 2	5001	Link Object 2
.....	Increased number in turn	
FRT Link Object 1	7000	FRT Link Object 1
FRT Link Object 2	7001	FRT Link Object 2
.....	Increased number in turn	

6.2.1.2 FAL management objects

6.2.1.2.1 MOB header class

MOB Header Class represents the version number of MOB.

6.2.1.2.1.1 Formal model

MOB header class is presented as follows:

ASE:	FAL MANAGEMENT ASE
CLASS:	MOB HEADER
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m) Key Attribute:	Object ID
2. (m) Attribute:	MOB Revision Number
SERVICES:	
1. (o) OpsService:	Read

6.2.1.2.1.2 Attributes

Object ID

This attribute identifies MOB Header object in MOB. The value for a device MIB Header object is 1.

SMIB Revision Number

This attribute indicates the revision number of MOB which is assigned by user.

6.2.1.2.1.3 Services

Read

The service allows user to read MOB Header Class attributes.

6.2.1.2.2 Device descriptor class

This class specifies the basic attributes of a device, i.e. Device ID, Device Type, Device Tag, IP address, etc. Through FAL management services, the user application program can get or set the basic attributes of this device.

6.2.1.2.2.1 Formal model

ASE:		FAL MANAGEMENT ASE	
CLASS:		DEVICE DESCRIPTOR	
CLASS ID:		Not Used	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1.	(m)	Key Attribute:	Object ID
2.	(m)	Attribute:	Reserved
3.	(m)	Attribute:	Application Type
4.	(m)	Key Attribute:	Device ID
5.	(m)	Attribute:	PD_Tag
6.	(m)	Attribute:	Active IP Address
7.	(m)	Attribute:	Device Type
8.	(m)	Attribute:	Status
9.	(m)	Attribute:	Device Version
10.	(m)	Attribute:	Annunciation Interval
11.	(m)	Attribute:	Annunciation Version Number
12.	(m)	Attribute:	Device Redundancy State
13.	(m)	Attribute:	Device Redundancy Number
14.	(m)	Attribute:	LAN Redundancy Port
15.	(m)	Attribute:	Max Redundancy Number
16.	(m)	Attribute:	Duplicate Tag Detected
SERVICES:			
1.	(o)	OpsService:	EM_DetectingDevice
2.	(o)	OpsService:	EM_OnlineReply
3.	(o)	OpsService:	EM_ConfiguringDevice
4.	(o)	OpsService:	EM_GetDeviceAttribute
5.	(o)	OpsService:	EM_SetDefaultValue
6.	(o)	OpsService:	EM_ActiveNotification

6.2.1.2.2.2 Attributes

Object ID

This attribute identifies the Device Descriptor object in MOB. Its value is 2 for the Device Descriptor object.

Reserved

This field is reserved.

Application type

This attribute indicates the application type of local device. There are three options.

- 0: Regular ISO/IEC 8802-3 based applications
- 1: RT applications
- 2: FRT applications

Device ID

This attribute specifies the vendor specific physical Device. It is set by the manufacturer.

PD_Tag

This attribute specifies the site administered name (tag) assigned to the device.

Active IP address

This attribute indicates current operable IP address.

Device type

This attribute specifies the type of device and its capability. It is set by the manufacturer.

Status

This attribute indicates the status of local device. There are three options.

- 0: No address
- 1: Unconfigured
- 2: Configured or operable

Device version

This attribute is the version number of the device. It is assigned by the vendor when it is produced.

Annunciation interval

This attribute specifies the milliseconds between annunciation messages. Its default value is 15 000 (15 s).

Annunciation version number

This attribute is the version number that represents the composite state of the static information in the applications in the device. Each time there is change to the static data of any application in the device; this version number is incremented by 1. Its value initializes to zero if the device is initialized.

Device redundancy state

This attribute specifies the actual device redundancy state of the device. 0 indicates the device is active while 1 represents that the device is back-up. If the device is not participating in redundancy, its value is 0xFF.

Device redundancy number

This attribute specifies the device redundancy number. Once the active device fails, the minimum number of the back-up device firstly becomes active device.

Max redundancy number

This attribute specifies the max redundancy number.

LAN redundancy port

This attribute specifies the port which is used to exchange data between redundancy devices.

Duplicate tag detected

The attribute specifies whether the device PD_Tag conflict with other device PD_Tag in network or not. That value is TRUE means occurring conflict.

6.2.1.2.2.3 Services

EM_DetectingDevice

This optional service allow user to find device information according to device PD_Tag.

EM_OnlineReply

This optional service is used to reply the device which sends EM_DetectingDevice request.

EM_ConfiguringDevice

This optional service allows user to set the device attributes.

EM_GetDeviceAttribute

This optional service allows user to get the device attributes.

EM_SetDefaultValue

This optional service allows user to clear the device attributes.

EM_ActiveNotification

This service is used for a device to announce its presence on the network.

6.2.1.2.3 Time synchronization class

In order implemented the clock synchronization of device in network, each device maintains a local Current Time. Through time synchronization, the local current can be synchronized with the time server in requested synchronization precision.

6.2.1.2.3.1 Formal model

This class is presented as follows:

ASE:	FAL MANAGEMENT ASE
CLASS:	TIME SYNCHRONIZATION
CLASS ID:	Not Used
PARENT CLASS:	TOP

ATTRIBUTES:

1.	(m)	Key Attribute:	Object ID
2.	(m)	Attribute:	Reserved
3.	(m)	Attribute:	Primary Time Server
4.	(m)	Attribute:	Secondary Time Server
5.	(m)	Attribute:	Time Request Timeout
6.	(m)	Attribute:	Time Request Interval
7.	(m)	Attribute:	Capable Time Sync Class
8.	(m)	Attribute:	Target Time Sync Class
9.	(m)	Attribute:	Current Time
10.	(m)	Attribute:	Standard Time Difference

SERVICES:

1.	(o)	OpsService:	Read
2.	(o)	OpsService:	Write
3.	(o)	OpsService:	Clock Synchronization service defined in RFC 2030 or

IEC 61588.

6.2.1.2.3.2 Attributes

Object ID

This attribute identifies the Time Synchronization Object in MOB. Its value is 3 for Time Synchronization Object.

Reserved

This field is reserved.

Primary time server

This attribute specifies the IP address of Primary Time Server.

Secondary time server

This attribute specifies the IP address of Secondary Time Server.

Time request timeout

This attribute specifies the time (in microseconds) of time client to wait for the request from time server.

Time request interval

This attribute specifies the time interval (in seconds) of time client to transmit the synchronization request.

Capable time sync class

This attribute specifies time synchronization precision supported by the time client.

- 0: No precision requirement
- 1: Time Synchronization precision < 1 s
- 2: Time Synchronization precision < 100 ms
- 3: Time Synchronization precision < 10 ms
- 4: Time Synchronization precision < 1 ms
- 5: Time Synchronization precision < 100 μ s
- 6: Time Synchronization precision < 10 μ s
- 7: Time Synchronization precision < 1 μ s
- 8: Time Synchronization precision < 100 ns
- 9: Time Synchronization precision < 10 ns

Target time sync class

This attribute indicates the desired time synchronization precision supported by time client. It may be set by user application.

- 0: no precision requirement
- 1: Time Synchronization precision < 1 s;
- 2: Time Synchronization precision < 100 ms;
- 3: Time Synchronization precision < 10 ms;
- 4: Time Synchronization precision < 1 ms;
- 5: Time Synchronization precision < 100 μ s;
- 6: Time Synchronization precision < 10 μ s;
- 7: Time Synchronization precision < 1 μ s

8: Time Synchronization precision < 100 ns

9: Time Synchronization precision < 10 ns

Current time

This attribute indicates current clock time in local device calculated since 1 January , 1984. Its data type is TimeOfDay.

Standard time difference

This attribute indicates the time difference between standard system time and current time. Its data type is 8 octets PrecisionTimeDifference.

6.2.1.2.3 Services

Read

The service allows the user to read Clock Synchronization Object Class attributes.

Write

The service allows user to set Clock Synchronization Object Class attributes.

Clock synchronization service

The necessary services are defined in RFC 2030 or IEC 61588.

6.2.1.2.4 Confirmed service max response time class

6.2.1.2.4.1 Formal model

ASE:		FAL MANAGEMENT ASE
CLASS:		MAX RESPONSE TIME
CLASS ID:		Not Used
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Key Attribute: Object ID
2.	(m)	Attribute: Reserved
3.	(m)	Attribute: Max Response Time
SERVICES:		
1.	(o)	OpsService: Read
2.	(o)	OpsService: Write

6.2.1.2.4.2 Attributes

Object ID

This attribute identifies Confirmed Service Max Response Time Class Object in MOB. Its value is 4.

Reserved

This field is reserved.

Max response time

This attribute specifies max response time to wait for response since the request primitive is sent for a confirmed service. Its data type is 8 octets of PrecisionTimeDifference.

6.2.1.2.4.3 Services

Read

The service allows user to read the attributes of confirmed service max response time object.

Write

The service allows user to write the attributes of confirmed service max response time object.

6.2.1.2.5 Communication scheduling management class

This class specifies the information used in an ECSME.

6.2.1.2.5.1 Formal model

ASE:		FAL MANAGEMENT ASE
CLASS:		COMMUNICATION SCHEDULE MANAGEMENT
CLASS ID:		Not Used
PARENT CLASS:		TOP
ATTRIBUTES:		
1. (m)	Key Attribute:	Object ID
2. (m)	Attribute:	Communication Macrocycle
3. (m)	Attribute:	NonPeriodic Data Transfer Offset
4. (m)	Attribute:	Communication Macrocycle Version Number
SERVICES:		
1. (o)	OpsService:	Read
2. (o)	OpsService:	Write

6.2.1.2.5.2 Attributes

Object ID

This attribute identifies the Communication Scheduling Management Object in the MOB. Its value is 5.

Communication macrocycle

The data type of Communication Macrocycle is 8 octets of PrecisionTimeDifference or PrecisionTimeDifference and may be set by the user. The default value of 0xFFFFFFFFFFFFFFFF indicates that Communication Macrocycle has not been configured.

Non-periodic data transfer offset

This attribute specifies time offset of the NonPeriodicDataTransfer starting time from the starting time of a communication macrocycle. Its data type is 8 octets of PrecisionTimeDifference. The default value of 0xFFFFFFFFFFFFFFFF indicates that this attribute has not been configured.

Communication cycle version number

This attribute specifies the configuration state of a Communication Scheduling Management object. Each time the object is modified, the value of this attribute is incremented by 1.

6.2.1.2.5.3 Services

Read

The service allows user to read Communication Cycle Management Object Class attributes.

Write

The service allows user to set Communication Cycle Management Object Class attributes.

6.2.1.2.6 Device application information class

6.2.1.2.6.1 Formal model

ASE:	FAL MANAGEMENT ASE
CLASS:	DEVICE APPLICATION INFORMATION
CLASS ID:	Not Used
PARENT CLASS:	TOP

ATTRIBUTES:

- | | | | |
|----|-----|----------------|--------------|
| 1. | (m) | Key Attribute: | Object ID |
| 2. | (m) | Attribute: | XDDL Version |

SERVICES:

- | | | | |
|----|-----|-------------|------|
| 1. | (o) | OpsService: | Read |
|----|-----|-------------|------|

6.2.1.2.6.2 Attributes**Object ID**

This attribute identifies the Device Application Information object in MOB. Its value is 6.

XDDL Version

This attribute indicates the version of XDDL document. Its data type is Unsigned16, where the higher octet specifies primary version number while the lower octet indicates the secondary version number.

6.2.1.2.6.3 Services**Read**

The service allows user to read Device Application Information Object Class attributes.

6.2.1.2.7 FB application information header class**6.2.1.2.7.1 Formal model**

ASE:	FAL MANAGEMENT ASE
CLASS:	FB APPLICATION INFORMATION HEADER
CLASS ID:	Not Used
PARENT CLASS:	TOP

ATTRIBUTES:

- | | | | |
|----|-----|----------------|---|
| 1. | (m) | Key Attribute: | Object ID |
| 2. | (m) | Attribute: | Number of FB Application Information Object |
| 3. | (m) | Attribute: | First Number of FB Application Information Object |

SERVICES:

- | | | | |
|----|-----|-------------|------|
| 1. | (o) | OpsService: | Read |
|----|-----|-------------|------|

6.2.1.2.7.2 Attributes**Object ID**

This attribute identifies the FB Application Information Object Header object in MOB. Its value is 7.

Number of FB application information object

This attribute indicates the number of FB Application Information Object in local device.

First Number of FB application information object

This attribute indicates the first number of FB Application Information Object in MOB. Its value is greater than ObjectID of the domain application object header.

6.2.1.2.7.3 Services**Read**

The optional service allows user to read the attributes of FB Application Information Object Header class.

6.2.1.2.8 Link object header class**6.2.1.2.8.1 Formal model**

ASE:	FAL MANAGEMENT ASE
CLASS:	LINK OBJECT HEADER

CLASS ID:		Not Used
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Key Attribute: Object ID
2.	(m)	Attribute: Number of Link Object
3.	(m)	Attribute: First Number of Link Object
4.	(m)	Attribute: Number of Configured Link Object
5.	(m)	Attribute: Number of UnConfigured Link Object
SERVICES:		
1.	(o)	OpsService: Read

6.2.1.2.8.2 Attributes

Object ID

This attribute identifies the Link Object Header object in MOB. Its value is 8.

Number of link object

This attribute indicates the number of link objects in MOB.

First number of link object

This attribute indicates the first number of link object in MOB.

Number of configured link object

This attribute indicates the number of link objects configured by users.

Number of unconfigured link object

This attribute indicates the number of unconfigured link objects in local device.

6.2.1.2.8.3 Service

Read

The optional service allows user to read the attributes of Link Object Header Class.

6.2.1.2.9 Domain application information header class

6.2.1.2.9.1 Formal model

ASE:		FAL MANAGEMENT ASE
CLASS:		DOMAIN APPLICATION INFORMATION HEADER
CLASS ID:		Not Used
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Key Attribute: Object ID
2.	(m)	Attribute: Number of Domain Application Object
3.	(m)	Attribute: First Number of Domain Application Object
4.	(m)	Attribute: Number of Configured Domain Object
5.	(m)	Attribute: Number of UnConfigured Domain Object
SERVICES:		
1.	(o)	Ops Service: Read

6.2.1.2.9.2 Attributes

Object ID

This attribute identifies domain Application Information Object Header Class in MOB. Its value is 9.

Number of somain application informationobject

This attribute indicates the number of Domain Application Information Object in local device.

First number of domain application object

This attribute indicates the first number of Domain Application Information Object in local device.

Number of configured domain object

This attribute indicates the number of configured Domain objects in local device.

Number of unconfigured domain object

This attribute indicates the number of unconfigured Domain objects in local device.

6.2.1.2.9.3 Services

Read

This optional service provides reading the attribute of Domain Application Object Header Class.

6.2.1.2.10 FRT Link object header class

6.2.1.2.10.1 Formal model

ASE:	FAL MANAGEMENT ASE
CLASS:	FRT LINK OBJECT HEADER
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Key Attribute: Object ID
2. (m)	Attribute: Number of FRT Link Object
3. (m)	Attribute: First Number of FRT Link Object
4. (m)	Attribute: Number of Configured FRT Link Object
5. (m)	Attribute: Number of UnConfigured FRT Link Object
SERVICES:	
1. (o)	OpsService: FRT Read

6.2.1.2.10.2 Attributes

Object ID

This attribute identifies the Link Object Header object in MOB. Its value is 8.

Number of FRT link object

This attribute indicates the number of FRT link objects in MOB.

First number of FRT link object

This attribute indicates the first number of FRT link object in MOB.

Number of configured FRT link object

This attribute indicates the number of FRT link objects configured by users.

Number of unconfigured FRT link object

This attribute indicates the number of unconfigured FRT link objects in local device.

6.2.1.2.10.3 Service

FRTRead

The optional service allows user to read the attributes of FRT Link Object Header Class.

6.2.1.2.11 FB application information class

6.2.1.2.11.1 Formal model

ASE:	FAL MANAGEMENT ASE
CLASS:	FB APPLICATION INFORMATION
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Key Attribute: Object ID
2. (m)	Attribute: Reserved
3. (m)	Attribute: FB Name
4. (m)	Attribute: FB Type
5. (m)	Attribute: Max Number of Instantiation
6. (m)	Attribute: FB Execution Time
7. (m)	Attribute: First Number of Instantiation
SERVICES:	
1. (o)	Ops Service: Read

6.2.1.2.11.2 Attributes

Object ID

This attribute identifies FB Application Information Object in MOB. The number of ObjectID of FB Application Information Objects should be appointed in series.

Reserved

This attribute is a reserved field.

FB name

This attribute is used to identify an FB in local device.

FB type

This attribute indicates the type of an FB.

Max number of instantiation

This attribute indicates the max instantiation number of an FB.

FB execution time

This attribute specifies the execution time (in milliseconds for RT applications and microseconds for FRT applications) of a FB.

First Number of Instantiation

This attribute defines the first number which can be assigned for FB instantiation.

6.2.1.2.11.3 Services

Read

This optional service provides reading the attributes of FB Application Information Object.

6.2.1.2.12 Link object class

6.2.1.2.12.1 Formal model

ASE:	FAL MANAGEMENT ASE
CLASS:	LINK OBJECT
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Key Attribute: Object ID
2. (m)	Attribute: LocalAppID

3.	(m)	Attribute:	Local Object ID
4.	(m)	Attribute:	RemoteAppID
5.	(m)	Attribute:	RemoteObjectID
6.	(m)	Attribute:	ServiceOperation
7.	(m)	Attribute:	ServiceRole
8.	(m)	Attribute:	RemoteIPAddress
9.	(m)	Attribute:	SendTimeOffset

SERVICES:

1.	(o)	Ops Service:	Read
2.	(o)	Ops Service:	Write

6.2.1.2.12.2 Attributes**Object ID**

This attribute identifies Link Object in MOB. The number of ObjectID of Link Object should be appointed in series.

Local app ID

This attribute identifies the local FB instance.

Local object ID

This attribute identifies the local variant object.

Remote app ID

This attribute identifies the remote FB instantiation.

RemoteObjectID

This attribute identifies the remote variant object.

ServiceOperation

This attribute specifies the application service to be used in the relevant communication relationship.

- 0: local link, no application service is used
- 1 through 17: the ServiceID of the Type 14 application services is used
- Others: invalid service

ServiceRole

This attribute defines the AREP role of local device in communication process.

- 0: SENDER, indicating that the AREP role of the local device is CLIENT or PUBLISHER
- 1: RECEIVER, indicating that the AREP role of the local device is SERVER or SUBSCRIBER
- Others: Link Object is invalid, and 0xFF indicates that the Link Object is not configured or the Link Object has been deleted

RemoteIPAddress

This attribute identifies IP address of remote device. This attribute can be ignored if local FB instantiation object and remote FB instantiation object are in the same device.

SendTimeOffset

This attribute defines the time offset when the relevant message should be sent from the start time of a communication macrocycle. This attribute is valid when Service ID is 0x0e (VARIABLEDISTRIBUTE) and ServiceRole is 0.

6.2.1.2.12.3 Service

Read

This optional service permits users to read the attributes of Link Object.

Write

This optional service permits users to configure the attributes of Link Object.

6.2.1.2.13 Domain application information class

6.2.1.2.13.1 Formal model

ASE:	FAL MANAGEMENT ASE
CLASS:	DOMAIN APPLICATION INFORMATION
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Key Attribute: Object ID
2. (m)	Attribute: Domain Object ID
3. (m)	Attribute: ConfigurationStatus
4. (m)	Attribute: Reserved
5. (m)	Attribute: Domain Name
SERVICES:	
1. (o)	Ops Service: Read

6.2.1.2.13.2 Attributes

Object ID

This attribute identifies Domain Application Information Object in MOB. The number of ObjectID of Domain Application Information Objects should be appointed in series.

Domain object ID

This attribute indicates the index of a Domain Object.

ConfigurationStatus

This attribute indicates the configuration status of a Domain Object. It is TRUE if Domain Object has been configured.

Reserved

This attribute is a reserved field.

Domain name

This attribute identifies a local domain.

6.2.1.2.13.3 Services

Read

This optional service permits users to read the attributes of Domain Application Information Object.

6.2.1.2.14 FRT Link object class

6.2.1.2.14.1 Formal model

ASE:	FAL MANAGEMENT ASE
CLASS:	FRT LINK OBJECT
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	

1.	(m)	Key Attribute:	Object ID
2.	(m)	Attribute:	Local Object ID
3.	(m)	Attribute:	RemoteObjectID
4.	(m)	Attribute:	ServiceOperation
5.	(m)	Attribute:	ServiceRole
6.	(m)	Attribute:	RemoteMACAddress
7.	(m)	Attribute:	SendTimeOffset
8.	(m)	Attribute:	ValidBitOffset
9.	(m)	Attribute:	ValidBitNumber

SERVICES:

1.	(o)	Ops Service:	FRTRead
2.	(o)	Ops Service:	FRTWrite

6.2.1.2.14.2 Attributes**Object ID**

This attribute identifies FRT Link Object in MOB. The number of ObjectID of FRT Link Object should be appointed in series.

Local object ID

This attribute identifies the local variant object.

RemoteObjectID

This attribute identifies the remote variant object.

ServiceOperation

This attribute specifies the application service to be used in the relevant communication relationship.

- 0: local link, no application service is used
- 1 through 17: the ServiceID of the Type 14 application services is used
- Others: invalid service

ServiceRole

This attribute defines the AREP role of local device in communication process.

- 0: SENDER, indicating that the AREP role of the local device is CLIENT or PUBLISHER
- 1: RECEIVER, indicating that the AREP role of the local device is SERVER or SUBSCRIBER
- Others: Link Object is invalid, and 0xFF indicates that the Link Object is not configured or the Link Object has been deleted

RemoteMACAddress

This attribute identifies MAC address of remote device. This attribute can be ignored if local FB instantiation object and remote FB instantiation object are in the same device.

SendTimeOffset

This attribute defines the time offset when the relevant message should be sent from the start time of a communication macrocycle. This attribute is valid when Service ID is 0x12 (FRTVARIABLEDISTRIBUTE) and ServiceRole is 0.

ValidBitOffset

This attribute defines the bit offset when the relevant message should be sent or received from the start time of field of DATA in FRTVARIABLEDISTRIBUTE Service. This attribute is valid when Service ID is 0x12 (FRTVARIABLEDISTRIBUTE).

ValidBitNumber

This attribute defines the bit number when the relevant message should be sent or received from the start time of field of DATA in FRTVARIABLEDISTRIBUTE Service. This attribute is valid when Service ID is 0x12 (FRTVARIABLEDISTRIBUTE).

6.2.1.2.14.3 Service

FRTRead

This optional service permits users to read the attributes of FRT Link Object.

FRTWrite

This optional service permits users to configure the attributes of FRT Link Object.

6.2.2 FAL application access entity ASE

6.2.2.1 Common access-related attributes

Password

This attribute specifies the password for the access right.

Access groups

This attribute allows restricting the access only to clients who belong to one of the groups defined by the attribute.

This attribute specifies if the object belongs to a group of users, as shown in Table 21. If one of the bits is set to 1, it indicates a group number to which the object belongs.

Table 21 – Access group assignment

Bit	Signification
7	Access Group 1
6	Access Group 2
5	Access Group 3
4	Access Group 4
3	Access Group 5
2	Access Group 6
1	Access Group 7
0	Access Group 8

Access rights

This attribute defines the type of access defined for the domain, as show in Table 22. The corresponding access right is permitted while related bit has been set.

Table 22 – Access rights assignment

Bit	Name	Signification
7	R	Right to Read for the registered Password
6	W	Right to Write for the registered Password
5	U	Right to Use the registered Password
3	Rg	Right to Read for the Access Groups
2	Wg	Right to Write for the Access Groups
1	Ug	Right to Use the Access Groups

6.2.2.2 Domain object**6.2.2.2.1 Overview**

A domain represents a memory area whose contents may be data or program, and the memory area is represented as an ordered sequence of octets. A domain can be uploaded or downloaded, to maintain integrity, only one download or upload operation for a domain is permitted at the same time.

The domain ASE provides the following services shown in Table 23.

Table 23 – Services for domain object

Service name	Service	Type	Object
DomainDownload	Download service for Domain	Confirmed	Domain Object
DomainUpload	Upload service for Domain	Confirmed	

6.2.2.2.2 Formal model**ASE:****CLASS:****CLASS ID:****PARENT CLASS:****ATTRIBUTES:**

- | | | | |
|-----|-----|----------------|--------------------------|
| 1. | (m) | Key Attribute: | Object ID |
| 2. | (m) | Key Attribute: | Domain Name |
| 3. | (m) | Attribute: | Max Octets |
| 4. | (m) | Attribute: | Password |
| 5. | (m) | Attribute: | Access Groups |
| 6. | (m) | Attribute: | Access Rights |
| 7. | (m) | Attribute: | Local Address |
| 8. | (m) | Attribute: | Domain State |
| 9. | (m) | Attribute: | Last State |
| 10. | (m) | Attribute: | Used Application Counter |

SERVICES:

- | | | | |
|----|-----|--------------|----------------|
| 1. | (o) | Ops Service: | DomainDownload |
| 2. | (o) | Ops Service: | DomainUpload |

6.2.2.2.3 Attributes**Object ID**

This attribute is the object identifier.

Domain name

This attribute specifies the name of the domain object.

Max octets

This attribute specifies the maximum size of the domain in octets.

Password

See 6.2.2.1.

Access groups

See 6.2.2.1.

Access rights

See 6.2.2.1.

Local address

This attribute is a locally significant address of domain. The value 0xFFFFFFFF indicates that no local address is available.

Domain state

This attribute specifies the following states of domain.

- 0: EXISTENT
- 1: DOWNLOADING
- 2: UPLOADING
- 3: READY
- 4: IN-USE

Last state

This attribute specifies the state of domain before uploading or downloading.

Used application counter

This attribute specifies the number of the program using this domain. If the value of this attribute is more than 0, it is indicated that the domain is being used. The domain can not be re-downloaded while it is being used.

6.2.2.2.4 Services

DomainDownload

This optional service permits the client to download a domain.

DomainUpload

This optional service permits the client to upload a domain.

6.2.2.3 Report object

6.2.2.3.1 Overview

The report object is mainly used for the user to send important messages from one to some other devices (or in broadcast mode, to all other devices). It is the user who defines the conditions that triggers the event. The application process invokes the EventReport service to notify the event when the conditions are met. It is also the user who confirms the event.

Through the Event Notification service, the user transmits the Event Number and Event Data, e.g. measurement value, states and unit. The management of the Event Number is completed

by the user. The Event Notification may represent a collective alarm, where the data may contain information about which channel has triggered the collective alarm.

The receiver of the Event Notification may acknowledge the Event by using the AcknowledgeEventReport service. The counter value is transmitted with the AcknowledgeEventReport service. The counter serves to correlate the EventReport and the AcknowledgeEventReport.

The receiver of the EventReport may lock or unlock the EventReport using the ReportConditionChanging service.

The event ASE provides three services of event management, as shown in Table 24.

Table 24 – Service for report object

Service name	Service	Confirmed/ Unconfirmed	Object
Event Notification	Event Notification	Unconfirmed	ReportObject
AcknowledgeEventReport	Event Acknowledge	Confirmed	
Alter Event Condition Monitor	Alter Event Condition Monitor	Confirmed	

6.2.2.3.2 Formal model

ASE:

CLASS:

CLASS ID:

PARENT CLASS:

ATTRIBUTES:

- | | | | |
|----|-----|----------------|---------------|
| 1. | (m) | key attribute: | Object ID |
| 2. | (m) | attribute: | Length |
| 3. | (m) | attribute: | Password |
| 4. | (m) | attribute: | Access Groups |
| 5. | (m) | attribute: | Access Rights |
| 6. | (m) | attribute: | Local Address |
| 7. | (m) | attribute: | Enabled |

SERVICES:

- | | | | |
|----|-----|--------------|-------------------------|
| 1. | (o) | Ops Service: | EventReport |
| 2. | (o) | Ops Service: | AcknowledgeEventReport |
| 3. | (o) | Ops Service: | ReportConditionChanging |

6.2.2.3.3 Attributes

Object ID

This attribute indicates the identifier of the event object.

Length

This attribute indicates the max data length of event.

Password

See 6.2.2.1.

Access groups

See 6.2.2.1.

Access rights

See 6.2.2.1.

Local address

The pointer of the specific object, it may be used internally for addressing the object. If it is not needed, the value should be set as 0xFFFFFFFF.

Enabled

This attribute indicates the state of event object that is defined by State-Machine.

Enabled = TRUE ⇔ UNLOCKED

Indicates event object is unlocked and can be sent.

Enabled = TRUE ⇔ LOCKED

Indicates event object is locked and cannot be sent.

6.2.2.3.4 Services

EventReport

This optional service allows the server to notify one or more event.

AcknowledgeEventReport

This optional service enables a Client to acknowledge several event occurrences.

ReportConditionChanging

This optional service enables user to lock or unlock a event object.

6.2.2.4 Variable objects

6.2.2.4.1 Simple variable object

6.2.2.4.1.1 Formal model

ASE:	FAL Application ASE
CLASS:	SIMPLE VARIABLE
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	key attribute: Object ID
2. (m)	attribute: Data type
3. (m)	attribute: Length
4. (m)	attribute: Local Address
5. (m)	attribute: Password
6. (m)	attribute: Access Groups
7. (m)	attribute: Access Rights
SERVICES:	
1. (o)	Ops Service: Read
2. (o)	Ops Service: Write
3. (o)	Ops Service: VariableDistribute

6.2.2.4.1.2 Attributes

Object ID

This attribute indicates the identifier of variable object.

Data type

This attribute indicates the type of variable data.

Length

This attribute indicates the length of variable data.

Local address

This attribute is a locally significant address of variable object. The value of 0xFFFFFFFF indicates that no local address is available.

Password

See 6.2.2.1.

Access groups

See 6.2.2.1.

Access rights

See 6.2.2.1.

6.2.2.4.1.3 Services**Read**

This optional service may be used to read value of variable object.

Write

This optional service may be used to update value of variable object.

VariableDistribute

This optional service may be used to VariableDistribute value of variable.

6.2.2.4.2 Array variable class specification**6.2.2.4.2.1 Formal model**

ASE:	FAL Application ASE
CLASS:	ARRAY VARIABLE
CLASS ID:	Not Used
PARENT CLASS:	SIMPLE VARIABLE
ATTRIBUTES:	

1.	(m)	key attribute:	Object ID
2.	(m)	attribute:	Number of Elements

SERVICES:	
1.	(o) Ops Service: Read
2.	(o) Ops Service: Write
3.	(o) Ops Service: VariableDistribute

6.2.2.4.2.2 Attributes**Object ID**

This attribute indicates the identifier of variable object.

Number of elements

This attribute specifies the number of array elements for variable objects that are defined as array.

6.2.2.4.2.3 Services**Read**

This optional service may be used to read values of variable object.

Write

This optional service may be used to update values of variable object.

VariableDistribute

This optional service may be used to VariableDistribute value of variable.

6.2.2.4.3 Structure variable class specification

6.2.2.4.3.1 Formal model

ASE:		FAL Application ASE
CLASS:		STRUCTURE VARIABLE
CLASS ID:		Not Used
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	key attribute: Object ID
2.	(m)	attribute: Data type
3.	(m)	attribute: Length
4.	(m)	attribute: Local Address
5.	(m)	attribute: Password
6.	(m)	attribute: Access Groups
7.	(m)	attribute: Access Rights
SERVICES:		
1.	(o)	Ops Service: Read
2.	(o)	Ops Service: Write
3.	(o)	Ops Service: VariableDistribute

6.2.2.4.3.2 Attributes

Object ID

This attribute indicates the identifier of variable object.

Data type

This attribute specifies the data type of variable.

Length

This attribute is the length of variable data.

Local address

This attribute is a locally significant address of variable object. The value of 0xFFFFFFFF indicates that no local address is available.

Password

See 6.2.2.1.

Access groups

See 6.2.2.1.

Access rights

See 6.2.2.1.

6.2.2.4.3.3 Services

Read

This optional service may be used to read the value of variable object.

Write

This optional service may be used to update the value of variable object.

VariableDistribute

This optional service may be used to distribute the value of variable object.

6.2.3 Socket mapping ASE**6.2.3.1 Overview**

The socket mapping ASE (ESM ASE) is the application service element of Socket Mapping Entity. It provides the mapping between the application services and the lower layer. Its main functions are as follows.

- Mapping the application layer PDU to UDP protocol;
- Delivering the Application Layer PDUs into corresponding unsent buffer or queue according to the ServiceID identifier encapsulated in the PDUs.
- Providing overtime diagnosis and control for the confirmed services and returning positive or negative response to the corresponding entities
- Providing priority management of application layer services.
- Providing monitoring of the status of Link and reporting it to users.

6.2.3.2 Notional message transmitting management procedure [INFORMATIVE]

NOTE The following description is one means that a concrete protocol might use to implement these abstract services. See 1.2.

When receiving PDUs from the application access entity and the AL Management entity, the socket mapping entity firstly pushes it into corresponding queues according to the transmitting priority.

For the confirmed service PDUs, the socket mapping entity creates a timer object according to its ServiceID and assigns it an InstanceID. It will start the timer when the PDU is sent to the network by the ECSME and maintain it as follows.

- a) If the positive response is received before the Max Response Time of the confirmed message, the socket mapping entity delivers Result (+) response with relevant data to corresponding application layer entity, and then deletes the timer object.
- b) If the negative response is received before the Max Response Time of the confirmed message, the socket mapping entity delivers Result (–) response with error code to corresponding application layer entity, and then deletes the timer object.
- c) If no response is received when the timer runs over, the socket mapping entity delivers a negative response with Error class of excess time to corresponding application layer entity, and then deletes the timer object.

6.2.3.3 Notional message receiving management procedure [INFORMATIVE]

NOTE The following description is one means that a concrete protocol might use to implement these abstract services. See 1.2.

When receiving PDUs from the application access entity and the AL Management entity, the After powered-on and initialization, the device starts monitoring its communication port. When a message is received, the local socket mapping entity will deal with it according to the ServiceID encapsulated in it:

- a) If the message is received from the AL Management port, the socket mapping entity will deal with it according to the Service ID. If the Service ID is invalid, the message will be discarded. Otherwise

- 1) if the message is a request from remote devices, it will be delivered to the AL Management entity. If it is a confirmed service request, the local AL Management entity will give a response;
 - 2) if the message is the response for the local request, the socket mapping entity will deliver it to the local AL Management entity and delete the corresponding timer object.
- b) If the message is received from the Application Access port, the socket mapping entity will deal with it according to the Service ID. If the Service ID is invalid, the message will be discarded. Otherwise
- 1) if the message is a request from remote devices, it will be delivered to the application access entity. If it is a confirmed service request, the local application access entity will give a response;
 - 2) if the message is the response for the local request, the socket mapping entity will deliver it to the local application access entity and delete the corresponding timer object.

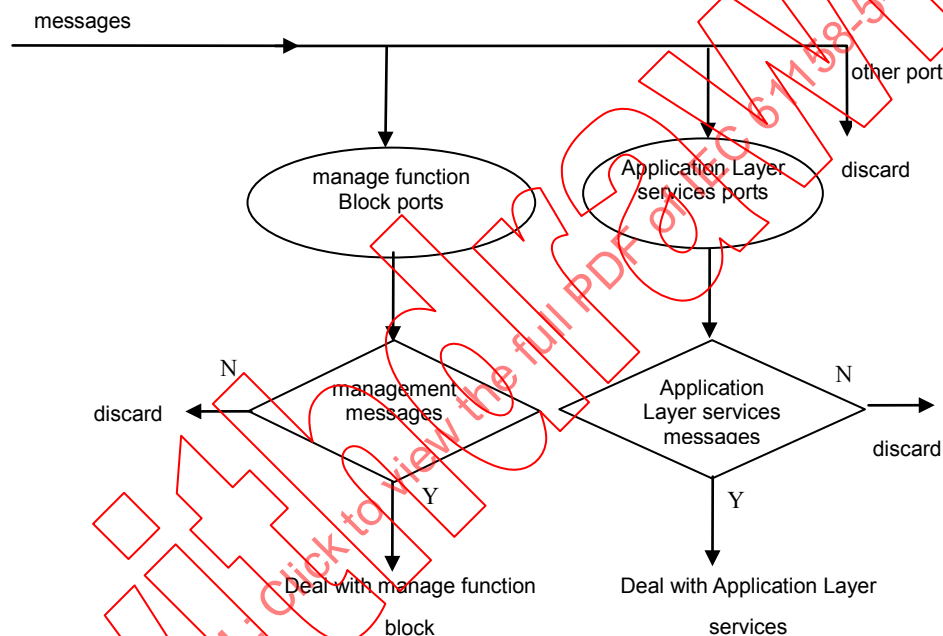


Figure 2 – Received message processing procedure

6.2.3.4 Socket mapping model class specification

6.2.3.4.1 Socket mapping manager class

6.2.3.4.1.1 Formal model

ASE:		SOCKET MAPPING ASE
CLASS:		SOCKET MAPPING
CLASS ID:		Not Used
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	attribute: Local IP Address
2.	(m)	attribute: Remote IP Address
3.	(m)	attribute: Active Udp Port
4.	(m)	attribute: Active Service ID
5.	(m)	attribute: Active Message Length
6.	(m)	attribute: Active Message ID
7.	(m)	attribute: Active Message Time
8.	(m)	attribute: Active Data Pointer
9.	(m)	attribute: Max Message Length
10.	(m)	attribute: Max Retransmit Number

6.2.3.4.1.2 Attributes**Local IP address**

This attribute indicates the IP Address of the local device.

Remote IP address

This attribute indicates the IP Address of the remote device.

Active UDP port

This attribute indicates the UDP Port to send messages.

Active service ID

This attribute indicates which service is used to send the message.

Active message length

This attribute indicates the length of active message.

Active message ID

This attribute indicates the identifier of active message.

Active message time

This attribute indicates the max. response time for active message. It should be set to 0 if it is unneeded.

Active data pointer

This attribute indicates the pointer to active message header.

Max message length

This attribute indicates the max. permitted length of message. It will refuse to send the message whose length is beyond the value of this attribute and return an error report.

Max retransmit number

This attribute indicates the max. times of retransmission.

6.2.3.4.1.3 Services

There are no operational services defined for this type of object.

6.2.3.4.2 Socket mapping timer class**6.2.3.4.2.1 Formal model**

ASE:	SocketMapping ASE
CLASS:	SocketTimer
CLASS ID:	Not Used
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	key attribute: TimerID
2. (m)	attribute: Active Service ID
3. (m)	attribute: Active Message ID

6.2.3.4.2.2 Attributes**TimerID**

This attribute indicates the identifier of the Timer.

Active service ID

This attribute indicates which service is used to send the message.

Active message ID

This attribute indicates the identifier of current message.

6.2.3.4.2.3 Services

There are no operational services defined for the type object.

6.2.4 Application layer communication services

6.2.4.1 FAL management service specification

FAL Management Entity provides the following six services shown in Table 25.

Table 25 – FAL management entity services

No.	service name	description
1	EM_DetectingDevice	This service is used to find the information of device. It is transferred by unicast or broadcast
2	EM_OnlineReply	This service is used as the reply to the EM_DetectingDevice service.
3	EM_ConfiguringDevice	This service is used to set the attribute of a device.
4	EM_GetDeviceAttribute	This service is used to read the attribute of a device.
5	EM_SetDefaultValue	This service is use to clear its configuration data.
6	EM_ActiveNotification	This service is used to periodically announce the presence of the device on the network

6.2.4.1.1 EM_DetectingDevice service

6.2.4.1.1.1 Overview

EM_DetectingDevice is an unconfirmed service. It is used to query the IP address of a device by PD_Tag to locate it on the network. This service is often sent using broadcast. A device may not receive any reply or receive several replies at the same time. The device receiving this query request primitive replies using EM_OnlineReply service.

An offline-configured device should query its PD_Tag using EM_DetectingDevice service to detect the PD_Tag conflict when it is connected on the network.

6.2.4.1.1.2 Service primitives

The service parameters for EM_DetectingDevice service are shown in Table 26.

Table 26 – EM_DetectingDevice service parameters

Parameter name	Req	Ind
Argument		
QueryType	M	M (=)
PD_Tag	M	M (=)
FB Tag	M	M (=)
Element ID	M	M (=)

Argument

The argument specifies the parameters of the service request.

QueryType

This parameter selects the following types to query:

- 0: PD_Tag query
- 1: FB Tag query
- 2: ElementID query

PD_Tag

This parameter specifies the Physical Device Tag.

FB tag

This parameter specifies the tag of a function block. It is required for FB tag queries.

Element ID

This parameter specifies a reference to the object of a FB parameter. The FB Tag shall also be present because ElementID is not unique outside of one FB.

6.2.4.1.1.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.1.2 EM_OnlineReply service**6.2.4.1.2.1 Service overview**

This service is used to send a reply to the initiator of an EM_DetectingDevice request. It returns Device ID and PD_Tag of the device queried.

6.2.4.1.2.2 Service primitives

The service parameters for EM_OnlineReply service are shown in Table 27.

Table 27 – EM_OnlineReply service parameters

Parameter name	Req	Ind
Argument		
Query Type	M	M (=)
Duplicate Tag Detected	M	M (=)
Queried Object IP Address	M	M (=)
Queried Object Device ID	M	M (=)
Queried Object PD_Tag	M	M (=)

Argument

The argument specifies the parameters of the service request.

QueryType

This parameter selects the type of query.

- 0: the following parameter specifies the information by PD_Tag query
- 1: the following parameter specifies the information by FB_Tag query
- 2: the following parameter specifies the information by ElementID query

Duplicate tag detected

This parameter describes the duplicated status of PD_Tag among devices.

Queried object IP address

This parameter specifies the IP address of the device queried.

Queried object device ID

This parameter specifies the ID of the device queried. Its length is 32 octets.

Queried object PD_Tag

This parameter specifies the PD_Tag of the device queried. Its length is 32 octets.

6.2.4.1.2.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.1.3 GetDeviceAttribute service

6.2.4.1.3.1 Service overview

EM_GetDeviceAttribute is a confirmed service. The host sends a request to get attributes of the device. After receiving EM_GetDeviceAttribute service, the device sends a Result(+) to the configuration application if performing normally, or else the AL Management entity sends a Result (–) to the configuration application.

6.2.4.1.3.2 Service primitives

The service parameters for EM_GetDeviceAttribute service are shown in Table 28.

Table 28 – EM_GetDeviceAttribute service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
Destination IP Address	M	M (=)		
Result (+)			S	S (=)
Device ID			M	M (=)
PD_Tag			M	M (=)
Status			M	M (=)
Device Type			M	M (=)
Annunciation Interval			M	M (=)
Annunciation Version Number			M	M (=)
Duplicate Tag Detected			M	M (=)
Device Redundancy Number			M	M (=)
LAN Redundancy Port			M	M (=)
Device Redundancy State			M	M (=)
Max Redundancy Number			M	M (=)
Active IP Address			M	M (=)
Result (–)			S	S (=)
Error Type			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

The argument specifies the parameters of the service request.

Destination IP address

This parameter specifies the destination IP address to which the service request is to be sent.

Result(+)

This selection type parameter indicates that the service request succeeded.

Device ID

This parameter specifies the identifier of the device.

PD_Tag

This parameter specifies the Physical Device Tag.

Status

This parameter specifies the following status of the device.

- 0: no address
- 1: unconfigured
- 2: configured

Device type

This parameter specifies the type of the device.

Annunciation interval

This parameter specifies the interval of sending the DeviceAnnunciation message.

Annunciation version number

This parameter specifies the version number of the message annunciated.

Duplicate tag detected

This parameter describes the duplicated status of PD_Tag among devices.

Device redundancy number

This parameter specifies the redundancy number of the device. Its value is 0 and the following parameters are invalid when the device is active.

LAN redundancy port

This conditional parameter specifies the value of the Port used to receive LAN Redundancy messages. It is present in the response primitive if the value of Redundancy Number is not 0.

Device redundancy state

This parameter specifies the following redundancy status of the device.

- 0: active status
- 1: redundancy status, it is present in the response primitive if the value of Redundancy Number is not 0

Max redundancy number

This parameter specifies the max redundancy number of the device. It is present in the response primitive if the value of Redundancy Number is not 0.

Active IP address

This parameter specifies the IP address of the active device. It is present in the response primitive if the value of Redundancy Number is not 0.

Result (–)

This selection type parameter indicates that the service request failed.

Error type

This parameter specifies the reason that caused failure.

6.2.4.1.3.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.1.4 DeviceAnnunciation service

6.2.4.1.4.1 Service overview

DeviceAnnunciation is an unconfirmed service. The device periodically sends this service request at the rate specified by Annunciation Interval to inform the configuration application. This service is often sent by broadcast.

6.2.4.1.4.2 Service primitives

The service parameters for EM_ActiveNotification service are shown in Table 29.

Table 29 – EM_ActiveNotification service parameters

Parameter name	Req	Ind
Argument		
Device ID	M	M (=)
PD_Tag	M	M (=)
Status	M	M (=)
Device Type	M	M (=)
Annunciation Version Number	M	M (=)
Device Redundancy Number	M	M (=)
Device Redundancy State	M	M (=)
LAN Redundancy Port	M	M (=)
Duplicate Tag Detected	M	M (=)
Max Redundancy Number	M	M (=)
Active IP Address	M	M (=)

Argument

The argument specifies the parameters of the service request.

Device ID

This parameter specifies the identifier of the device.

PD_Tag

This parameter specifies the Physical Device Tag.

Status

This parameter specifies the following status of the device.

- 0: no address
- 1: unconfigured
- 2: configured

Device type

This parameter specifies the type of the device. It is used to describe the functions of the device as defined by the manufacturer.

Annunciation version number

This parameter specifies the version number of the message annunciated.

Device redundancy number

This parameter specifies the redundancy number of the device.

LAN redundancy port

This conditional parameter specifies the value of the Port used to receive LAN Redundancy messages.

Duplicate tag detected

This parameter describes the duplicated status of PD_Tag among devices.

Max redundancy number

This parameter specifies the max redundancy number of the device.

Active IP address

This parameter specifies the IP address of active device.

6.2.4.1.4.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.1.5 EM_ConfiguringDevice service

6.2.4.1.5.1 Service overview

EM_ConfiguringDevice is a confirmed service which is sent using unicast. User application sends this service request to set the PD_Tag and others attributes of the device.

In order to avoid error, the DeviceID parameter within the request shall be equal to the DeviceID of the device. When executing, if the device has already has a PD_Tag, it shall clear its PD_Tag at first using EM_SetDefaultValue service.

6.2.4.1.5.2 Service primitives

The service parameters for EM_ConfiguringDevice service are shown in Table 30.

Table 30 – EM_ConfiguringDevice service primitives

Parameter name	Req	Ind	Rsp	Cnf
Argument				
Destination IP Address	M	M (=)		
Device ID	M	M (=)		
PD_Tag	M	M (=)		
Annunciation Interval	M	M (=)		
Duplicate Tag Detected	M	M (=)		
Device Redundancy Number	M	M (=)		
LAN Redundancy Port	M	M (=)		
Device Redundancy State	M	M (=)		
Max Redundancy Number	M	M (=)		
Active IP Address	M	M (=)		
Result (+)			S	S (=)
Max Redundancy Number			M	M (=)
Result (–)			S	S (=)
Error Type			M	M (=)

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.

Argument

The argument specifies the parameters of the service request.

Destination IP address

This parameter is the IP address to which the service request is to be sent.

Device ID

This parameter specifies the id of the device. Its length is 32 octets.

PD_Tag

This parameter specifies the Physical Device Tag. Its length is 32 octets.

Annunciation interval

This parameter specifies the interval of sending annunciation message. Its unit is second.

Duplicate tag detected

This parameter describes the duplicated status of PD_Tag among devices.

Device redundancy number

This parameter specifies the redundancy number of the device. Its value is 0 and the following parameters are invalid when the device is active.

LAN redundancy port

This conditional parameter specifies the value of the Port used to receive LAN Redundancy messages.

Device redundancy state

This parameter specifies the following redundancy status of the device.

0: active status

1: redundancy status, it is present in the response primitive if the value of Redundancy Number is not 0.

Max redundancy number

This parameter specifies the max redundancy number of the device. It is present in the response primitive if the value of Redundancy Number is not 0.

Active IP address

This parameter specifies the IP address of active device. It is present in the response primitive if the value of Redundancy Number is not 0.

Result(+)

This selection type parameter indicates that the service request succeeded.

Max redundancy number

This parameter specifies the max redundancy number of the device. It is present in the response primitive if the value of Redundancy Number is not 0.

Result (–)

This selection type parameter indicates that the service request failed.

Error type

This parameter specifies the reason that caused failure.

6.2.4.1.5.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.1.6 EM_SetDefaultValue service

6.2.4.1.6.1 Service overview

EM_SetDefaultValue is a confirmed service which is sent by unicast. User application sends this service request to clear the PD_Tag and set attributes of the device to default value. In order to avoid error, the parameter (DeviceID and PD_Tag) within the request shall be equal to the DeviceID and PD_Tag of the device.

6.2.4.1.6.2 Service primitives

The service parameters for EM_ConfiguringDevice service are shown in Table 31.

Table 31 – EM_SetDefaultValue service parameter

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M (=)		
Destination IP Address	M	M (=)		
Device ID	M	M (=)		
PD_Tag	M	M (=)		
Result(+)			S	S (=)
Result (–)			S	S (=)
Error Type			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

The argument specifies the parameters of the service request.

Destination IP address

This parameter is the IP address to which the service request is to be sent.

Device ID

This parameter specifies the id of the device. Its length is 32 octets.

PD_Tag

This parameter specifies the Physical Device Tag. Its length is 32 octets.

Result(+)

This selection type parameter indicates that the service request succeeded.

Result (–)

This selection type parameter indicates that the service request failed.

Error type

This parameter specifies the reason that caused failure.

6.2.4.1.6.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2 FAL application ASE specification**6.2.4.2.1 Domain download****6.2.4.2.1.1 Service overview**

Data and program can be downloaded by Domain Download service.

The following is the service procedure for Domain Downloading.

Step 1:

The user application examines at first whether the length of downloading data exceeds the max length (512 octets) or not. If it does, the application should divide the data into several consecutively numbered packets if the length of the data to be downloaded exceeds the max length.

Step 2:

User transmits the numbered data packet and the corresponding parameters by calling DomainDownload service.

Step 3:

After coding and packing, the Domain ASE transfers service package to the Socket Mapping ASE and sends it on the network by invoking UDP service.

Step 4:

After receiving package from UDP port, the device (the sever) decodes it and store the downloading data into local memory. Then, it returns a Result(+) response.

Step 5:

When receiving response, the user application will operate as follows:

- a) if receiving a Result(+) which indicates that the last downloading operation is successful. User application will repeat downloading operation using step 2 through step 5 till all Second if there remains other data to be downloaded. Otherwise, it ends the downloading operation;
- b) if receiving a Result (–), it will report the error to users.

6.2.4.2.1.2 Service primitives

The service parameters for the Domain Download service are shown in Table 32.

Table 32 – Parameters for domain download service

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M (=)		
SourceAppID	M	M (=)		
DestinationAppID	M	M (=)		
DestinationObject ID	M	M (=)		
DataNumber	M	M (=)		
MoreFollows	M	M (=)		
DataLength	M	M (=)		
LoadData	M	M (=)		
Result(+)			S	S (=)
Result (–)			S	S (=)
ErrorType			M	M (=)

NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.

Argument

The argument specifies the parameters of the service request.

SourceAppID

This parameter specifies the identifier of the source FB instantiation.

DestinationAppID

This parameter specifies the identifier of the destination FB instantiation.

Destination object ID

This parameter specifies the identifier of the domain object.

DataNumber

This parameter specifies the number of downloading data packet. It is numbered from 0.

MoreFollows

This parameter indicates whether there remains other data or not. When there remains other data to be downloaded, it is set to TRUE.

DataLength

This parameter specifies the length of the downloading data.

LoadData

This parameter specifies the downloading data.

Result (+)

This selection type parameter indicates that the service request succeeded.

Result (–)

This selection type parameter indicates that the service request failed.

Error Type

This parameter specifies the reason that caused failure.

6.2.4.2.1.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.2 Domain upload service

6.2.4.2.2.1 Service overview

Data and program can be uploaded by the Domain Upload service.

The following is the service procedure for Domain Uploading.

Step 1:

User application (client) transfers the uploading request with uploading domain object to the device (server) using DomainUpload service.

Step 2:

When receiving Domain Upload request from UDP port, the server will examine if the domain can be uploaded. If yes, it will compare the length of the uploading data excess with the max length (512 octets) of uploading data in a time. It will divide the uploading data into several packages numbered orderly if the length of uploading data exceeds with the max length.

Step3 :

The server transmits the numbered data packet and the corresponding parameters as response parameters.

Step 4:

When receiving response, the ASE operates as follows:

- a) If receiving a Result(+) response, it sends a Domain Upload request if the MoreFollows parameter in the response is TRUE. Otherwise, it sends a Domain Uploading operation;
- b) if receiving a Result (–) response, it reports the error to its AL-user.

6.2.4.2.2.2 Service primitives

The service parameters for Domain Upload service are shown in Table 33.

Table 33 – Parameters for domain upload service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
SourceAppID	M	M (=)		
DestinationAppID	M	M (=)		
DestinationObject ID	M	M (=)		
DataNumber	M	M (=)		
Result(+)			S	S (=)
MoreFollows			M	M (=)
DataLength			M	M (=)
LoadData			M	M (=)
Result (–)			S	S (=)
ErrorType			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1-2.				

Argument

The argument specifies the parameters of the service request.

SourceAppID

This parameter specifies the identifier of the source FB instantiation.

DestinationAppID

This parameter specifies the identifier of the destination FB instantiation.

Destination Object ID

This parameter specifies the identifier of the domain object.

DataNumber

This parameter specifies the number of service invoked during uploading. It is numbered from 0.

Result(+)

This selection type parameter indicates that the service request succeeded.

MoreFollows

This parameter indicates whether there remains other data or not. When there remains other data to be uploaded, it is set to TRUE.

DataLength

This parameter specifies the length of the uploading data.

LoadData

This parameter specifies the uploading data.

Result (–)

This selection type parameter indicates that the service request failed.

Error Type

This parameter specifies the reason that caused failure.

6.2.4.2.2.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.3 EventReport service**6.2.4.2.3.1 Service overview**

This service is used to transmit event notification. The event is transmitted by calling EventReport Service. It is an unconfirmed service using multicast or broadcast mode.

6.2.4.2.3.2 Service primitive

The service parameters for this service are shown in Table 34.

Table 34 – EventReport service parameters

Parameter name	Req	Ind
Argument		
DestinationAppID	M	M (=)
SourceAppID	M	M (=)
SourceObjectID	M	M (=)
EventNumber	M	M (=)
EventData	U	U (=)

Argument

The argument specifies the parameters of the service request.

DestinationAppID

This parameter is used to indicate the identifier of the destination application

SourceAppID

This parameter specifies the identifier of the source FB instantiation.

SourceObjectID

This parameter indicates the identifier of the object associated with this event.

EventNumber

The parameter indicates number of the event.

EventData

This parameter includes the specific data. The object data is mapped to the parameter data according to the description of the data type.

6.2.4.2.3.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.4 AcknowledgeEventReport service

6.2.4.2.4.1 Service overview

The service allows the user to acknowledge the Event Notification. It is a confirmed service. After the Event Notification is received, the user invokes this service to acknowledge the event.

6.2.4.2.4.2 Service primitive

The service parameters for this service are shown in Table 35.

Table 35 – AcknowledgeEventReport service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DestinationAppID	M	M (=)		
DestinationObjectID	M	M (=)		
EventNumber	M	M (=)		
Result(+)			S	S (=)
Result (–)			S	S (=)
ErrorType			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

The argument specifies the parameters of the service request.

DestinationAppID

This parameter is used to indicate the identifier of the destination application

DestinationObjectID

This parameter indicates the identifier of the object associated with this event.

EventNumber

The parameter indicates number of the event

Result(+)

This selection type parameter indicates that the service request succeeded.

Result (–)

This selection type parameter indicates that the service request failed.

Error Type

This parameter specifies the reason that caused failure.

6.2.4.2.4.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.5 ReportConditionChanging service

6.2.4.2.5.1 Service overview

ReportConditionChanging Service permits locking or unlocking the event object. The service is the confirmed service.

6.2.4.2.5.2 Service primitive

The service parameters for this service are shown in Table 36.

Table 36 – ReportConditionChanging service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DestinationAppID	M	M (=)		
DestinationObjectID	M	M (=)		
Enabled	M	M (=)		
Result(+)			S	S (=)
Result (-)			S	S (=)
Error Type			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

The argument specifies the parameters of the service request.

DestinationAppID

This parameter is used to indicate the identifier of the destination application

ObjectID

This parameter indicates the identifier of the object associated with this event.

Enabled

The boolean Parameter is used to set the value of the corresponding attribute of the event object.

Result(+)

This selection type parameter indicates that the service request succeeded.

Result (-)

This selection type parameter indicates that the service request failed.

ErrorType

This parameter specifies the reason that caused failure.

6.2.4.2.5.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.6 Read service

6.2.4.2.6.1 Service overview

This confirmed service may be used to read the value of a variable object. The value is transform between devices through peer-to-peer.

6.2.4.2.6.2 Service primitive

The service parameters for this service are shown in Table 37.

Table 37 – Read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DestinationAppID	M	M (=)		
DestinationObjectID	M	M (=)		
SubIndex	M	M (=)		
Result(+)			S	S (=)
Data			M	M (=)
Result (-)			S	S (=)
Error Type			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

This parameter carries the parameters of the service invocation.

DestinationAppID

This parameter specifies the identifier of the destination FB instantiation.

DestinationObjectID

This parameter indicates the identifier of the object associated with read object. The DestinationObjectID is 0 indicate that read all parameters of this FB instantiation.

SubIndex

This parameter indicates the subindex of the variable object. The value of 0 indicates that all elements of the object should be read.

Result(+)

This selection type parameter indicates that the service request succeeded.

Data

This parameter specifies the value to read. It shall accord with the data type of the read object.

Result (–)

This selection type parameter indicates that the service request failed.

ErrorType

This parameter specifies the reason that caused failure.

6.2.4.2.6.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.7 Write service

6.2.4.2.7.1 Service overview

This confirmed service is used to write the value of a variable object. The value is transform between devices through peer-to-peer.

6.2.4.2.7.2 Service primitive

The service parameters for this service are shown in Table 38.

Table 38 – Write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DestinationAppID	M	M (=)		
DestinationObjectID	M	M (=)		
SubIndex	M	M (=)		
Data	M	M (=)		
Result(+)			S	S (=)
Result (–)			S	S (=)
ErrorType			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

The argument specifies the parameters of the service request.

DestinationAppID

This parameter specifies the identifier of the destination FB instantiation.

DestinationObjectID

This parameter indicates the identifier of the variable object. The value of is 0 indicates that all parameters of this FB instantiation should be written.

SubIndex

This parameter indicates the sub-index of read object. If the SubIndex is 0 indicate that all element of the object be read.

Result(+)

This selection type parameter indicates that the service request succeeded.

Data

This parameter specifies the value to write.

Result (–)

This selection type parameter indicates that the service request failed.

ErrorType

This parameter specifies the reason that caused failure.

6.2.4.2.7.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.8 VariableDistribute service

6.2.4.2.8.1 Service overview

This service is used to transmit the specific value of Simple Variable, Array Variable and Record Variable. The main function of this service is to transmit input/output parameters among function blocks in the field devices, and it is an unconfirmed service. This service could be mapped into multicast for transmitting data to many devices.

6.2.4.2.8.2 Service primitive

The service parameters for this service are shown in Table 39.

Table 39 – VariableDistribute service parameters

Parameter name	Req	Ind
Argument		
Source AppID	M	M (=)
Source Object ID	M	M (=)
Data	M	M (=)

Argument

This parameter specifies the parameters of the service request.

Source appID

This parameter specifies the identifier of the source FB instantiation.

Source object ID

This parameter indicates the identifier of the object associated with VariableDistribute object.

Data

This parameter specifies the values to be VariableDistributed.

6.2.4.2.8.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.9 FRT Indication Distribute service

6.2.4.2.9.1 Service overview

This service is used to transmit the specific value of Simple Variable, Array Variable and Record Variable in FRT applications. The main function of this service is to transmit input/output parameters among function blocks in the field devices, and is also defined for a Type 14 device to broadcast all other devices if it has non-periodic data to be sent at the non-periodic data transferring phase.

When a Type 14 device has non-periodic data (such as alert, alarm, domain download/upload) to be sent, it shall broadcast the request primitive to all other nodes in the local micro-segment.

.It is an unconfirmed service. This service could be mapped into multicast for transmitting data to many devices.

6.2.4.2.9.2 Service primitive

The service parameters for this service are shown in Table 40.

Table 40 – FRTVariableDistribute service parameters

Parameter name	Req	Ind
Argument		
Source Object ID	M	M (=)
Data	M	M (=)

Argument

This parameter specifies the parameters of the service request.

Source object ID

This parameter indicates the identifier of the object associated with FRTVariableDistribute object.

Data

This parameter specifies the values to be FRTVariableDistributed.

6.2.4.2.9.3 Service procedure

The Unconfirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.10 FRTRead service

6.2.4.2.10.1 Service overview

This confirmed service may be used to read the value of a variable object in FRT applications. The value is transformed between devices through peer-to-peer.

6.2.4.2.10.2 Service primitive

The service parameters for this service are shown in Table 41.

Table 41 – FRTRead service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DestinationObjectID	M	M (=)		
SubIndex	M	M (=)		
Result(+)			S	S (=)
Data			M	M (=)
Result (–)			S	S (=)
Error Type			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

This parameter carries the parameters of the service invocation.

DestinationObjectID

This parameter indicates the identifier of the object associated with read object. The DestinationObjectID is 0 indicate that read all parameters of this FB instantiation.

SubIndex

This parameter indicates the subindex of the variable object. The value of 0 indicates that all elements of the object should be read.

Result(+)

This selection type parameter indicates that the service request succeeded.

Data

This parameter specifies the value to read. It shall accord with the data type of the read object.

Result (–)

This selection type parameter indicates that the service request failed.

ErrorType

This parameter specifies the reason that caused failure.

6.2.4.2.10.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.2.4.2.11 FRTWrite service

6.2.4.2.11.1 Service overview

This confirmed service is used to write the value of a variable object in FRT applications. The value is transform between devices through peer-to-peer.

6.2.4.2.11.2 Service primitive

The service parameters for this service are shown in Table 42.

Table 42 – FRTWrite service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DestinationObjectID	M	M (=)		
SubIndex	M	M (=)		
Data	M	M (=)		
Result(+)			S	S (=)
Result (–)			S	S (=)
ErrorType			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Argument

The argument specifies the parameters of the service request.

DestinationObjectID

This parameter indicates the identifier of the variable object. The value of is 0 indicates that all parameters of this FB instantiation should be written.

SubIndex

This parameter indicates the sub-index of read object. If the SubIndex is 0 indicate that all element of the object be read.

Result(+)

This selection type parameter indicates that the service request succeeded.

Data

This parameter specifies the value to write.

Result (–)

This selection type parameter indicates that the service request failed.

ErrorType

This parameter specifies the reason that caused failure.

6.2.4.2.11.3 Service procedure

The Confirmed Service Procedure specified in IEC/TR 61158-1, Clause 9 applies to this service.

6.3 Application relationship

6.3.1 General

In a VariableDistributed system, application processes communicate with each other so as to exchange application layer messages across well-defined application layer communications channels. These communication channels are abstracted in the AL as application relationships (ARs).

ARs are responsible for conveying messages between applications according to specific communication characteristics required by time-critical systems. Different combinations of these characteristics lead to the definition of different types of ARs. The characteristics of ARs are defined formally as attributes of AR Endpoint classes.

An implicit AR is defined at application layer, but explicit AR ASE entity is not defined. This implicit AR is used to establish all remote context relationship management and release communication channel.

6.3.2 AR endpoint

6.3.2.1 Overview

Each AP involved in an AR contains an endpoint of the AR. Each AR endpoint is defined within the AE of the AP. Each endpoint definition specifies a set of compatibility-related characteristics. These characteristics need to be configured appropriately for each endpoint for the AR to operate properly. The endpoint context is used by the AR ASE to manage the operation of the endpoint and the delivery of APDUs.

The messages that are conveyed by ARs are application layer service requests and responses. Each of these messages is submitted to the AR ASE for transfer by an AL ASE. Figure 3 illustrates this concept.

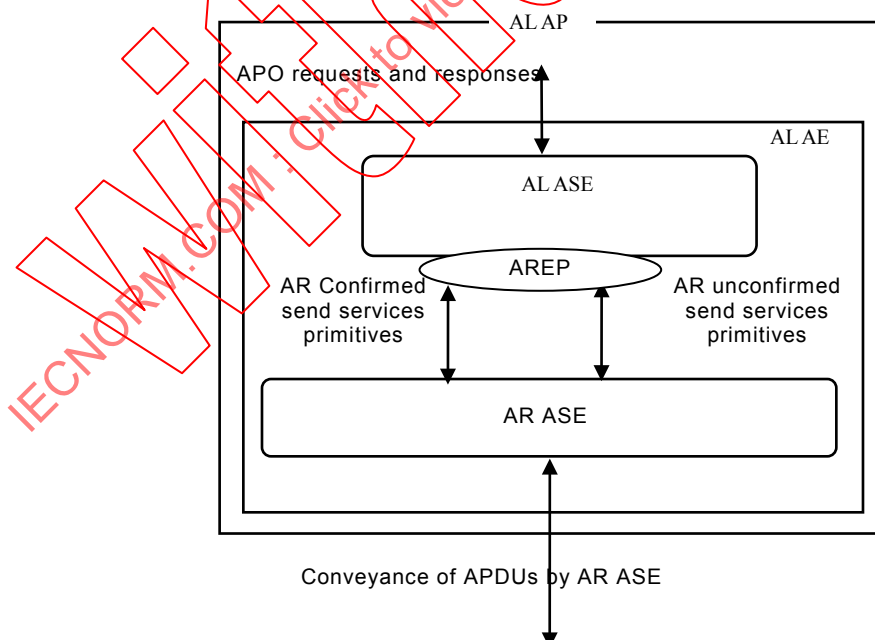


Figure 3 – AR ASE conveys APDUs between AP

6.3.2.2 AR endpoint role

The role of an AREP determines the permissible behavior of an AP at the AREP. An AREP role may be that of a client, server, peer (client and/or server), publisher and subscriber.

6.3.2.3 AR cardinality

From the point of view of a client or publisher endpoint, AR cardinality determines how many remote application processes are included in an AR. Cardinality is never expressed from the viewpoint of a server or a subscriber.

From the viewpoint of a client or peer endpoint, ARs are always 1-to-1. The client is never capable of issuing a request and waiting for responses from multiple servers.

From the viewpoint of a subscriber, ARs are 1-to-multiple. An AREP communicates with multiple or all devices' AREP.

6.3.2.4 Connection model

The AREP uses connectionless mode based on UDP/IP. This mode neither establishes nor releases connection. It is impossible to supervise connectionless AR.

6.3.2.5 Transfer type

The AREP support periodic or non-periodic data transferring modes.

a) Periodic data transferring

In this mode, application relationship is established after application process initiates a request. After that, it performs data transferring periodically.

b) Non-periodic data transferring

In this mode, only one data transferring is performed each time the application process request is sent.

c) Both

This mode supports both periodic and non-periodic mode.

6.3.2.6 Transfer feature

The AREP transfer data using either buffer or queue service.

6.3.2.7 Service type

The AREP support either confirmed or unconfirmed service.

6.3.2.8 Service relationship

The AREP supports both Client/Service and Publisher/Subscriber relationships.

6.3.2.9 AR establishment

The AREP uses locally established UDP channel to transfer APDU. It is established implicitly without any other services.

6.3.3 Application relationship endpoint class specification

6.3.3.1 Formal model

This model defines the common characteristics for all implicit AR endpoints. This implicit AR is not permitted to be written.

ASE:	AR ASE
CLASS:	AR ENDPOINT
CLASS ID:	Not Used
PARENT CLASS:	TOP
MANAGEMENT ATTRIBUTES:	

1.	(m)	key attribute:	Object ID
2.	(m)	attribute:	Local AP
3.	(m)	attribute:	Role
4.	(m)	attribute:	Initiator
5.	(m)	attribute:	TransferType
6.	(m)	attribute:	Service Type
7.	(m)	attribute:	Transfer Feature

6.3.3.2 Attributes

Object ID

This attribute indicates the identifier of the communication point.

Local AP

This attribute identifies the AP attached or configured to use the APEP using a local reference.

Role

This attribute specifies the role of AREP. Its value is shown as follows.

PEER -- It indicates whether the role of AREP is Client, Service, or Client and Service at the same time. If the role of local AREP is both Client and Server, it is specified that if local AREP is the initiator.

CLIENT – It indicates that the role of AREP is client.

SERVER – It indicates that the role of AREP is server.

PUBLISHER – It indicates that the role of AREP is publisher.

SUBSCRIBER – It indicates that the role of AREP is subscriber.

Initiator

This attribute indicates if the AREP is a communication initiator.

If Role is PEER, it is disabled.

If Role is Client or Publisher, it is set to TRUE.

If Role is Server or Subscriber, it is set to FALSE.

TransferType

This attribute indicates the transfer mode that the AREP supported.

PERIODIC – indicates the AREP support periodic data transfer service

NON PERIODIC – indicates the AREP support non-periodic data transfer service

BOTH -- indicates the AREP support both periodic and non-periodic data transfer service

ServiceType

This attribute indicates the service type that the AREP supported.

CONFIRMED – indicates that the AREP support confirmed data transfer service

UN CONFIRMED – indicates that the AREP support unconfirmed data transfer service

BOTH -- indicates that the AREP support both confirmed and unconfirmed data transfer service

Transfer feature

This attributes indicates the transferring type that the AREP supported.

QUEUE -- indicates the AREP support queued service transfer data

BUFFER – indicates the AREP support buffer service transfer data

6.3.3.3 Services

No service is defined.

6.4 Summary of application layer services

The summary of services supported in application layer is listed in Table 43.

Table 43 – Summary of application layer services

Index	Service name	ServiceID	Confirmed/ unconfirmed	Priority	Description of service
1	EM_DetectingDevice	1	Unconfirmed	2	Device Query
2	EM_OnlineReply	2	Confirmed	2	Device Query Reply
3	EM_GetDeviceAttribute	3	Confirmed	2	Get Device Attribute
4	EM_ActiveNotification	4	Unconfirmed	2	Device Annunciation
5	EM_ConfiguringDevice	5	Confirmed	2	Set Device Attribute
6	EM_SetDefaultValue	6	Confirmed	2	Clear Device Attribute
7	DomainDownload	10	Confirmed	2	Domain Download
8	DomainUpload	11	Confirmed	2	Domain Upload
9	Read	12	Confirmed	2	Variable read
10	Write	13	Confirmed	2	Variable write
11	VariableDistribute	14	Unconfirmed	0	Variable VariableDistribute
12	EventReport	15	Confirmed	1	Event notification
13	AcknowledgeEventReport	16	Confirmed	1	Confirmed Acknowledge Event Notification
14	ReportConditionChanging	17	Confirmed	1	Alter Event Condition Monitor
15	FRTVariableDistribute	18	Unconfirmed	0	FRT Variable VariableDistribute
16	FRTRead	19	Confirmed	2	FRT Variable read
17	FRTWrite	20	Confirmed	2	FRT Variable write
18	SNTP Service		Unconfirmed	3	
19	ARP Service			1	
20	RARP, ICMP, IGMP, DHCP Service			4	
21	FTP, TFTP, HTTP Services			5	

Bibliography

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*

IEC 61131-1, *Programmable controllers – Part 1: General information*

IEC/TR 61158-1:2007, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-3-14, *Industrial communication networks – Fieldbus specifications – Part 3-14: Data-link layer service definition – Type 14 elements*

IEC 61158-6-14, *Industrial communication networks – Fieldbus specifications – Part 6-14: Application layer protocol specification – Type 14 elements*

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO/IEC 8859-1, *Information technology – 8-bit single-byte coded graphic character sets – Part 1: Latin alphabet No. 1*

ISO/IEC 10646-1, *Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Architecture and Basic Multilingual Plane*

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

IEEE 802.11, *IEEE Standards for Information Technology – Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*

IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2010

SOMMAIRE

AVANT-PROPOS	96
INTRODUCTION	98
1 Domaine d'application	99
1.1 Vue d'ensemble	99
1.2 Spécifications	100
1.3 Conformité	100
2 Références normatives	100
3 Termes, définitions, symboles, abréviations et conventions	101
3.1 Termes de l'ISO/CEI 7498-1	101
3.2 Termes de l'ISO/CEI 8822	101
3.3 Termes de l'ISO/CEI 9545	101
3.4 Termes de l'ISO/CEI 8824-1	102
3.5 Définitions spécifiques à la couche application des bus de terrain	102
3.6 Symboles et abréviations	104
3.7 Conventions	106
4 Concepts	109
5 Data type ASE	109
5.1 Vue d'ensemble	109
5.2 Définition formelle des objets de types de données	109
5.3 Types de données définis pour la FAL	110
5.4 Spécification de service de l'ASE de type de données	132
6 Spécification de modèle de communication	132
6.1 Généralités	132
6.2 Les ASE	134
6.3 Relation d'applications	184
6.4 Résumé des services de couche application	188
Bibliographie	189
Figure 1 – Entité de couche application	133
Figure 2 – Procédure de traitement des messages reçus	159
Figure 3 – L'ASE d'AP achemine des APDU entre des AP	185
Tableau 1 – Attribut de variable normalisée à 2 octets	119
Tableau 2 – Codage de variable normalisée à 2 octets	119
Tableau 3 – Attribut de variable normalisée à 4 octets	119
Tableau 4 – Codage de variable normalisée à 4 octets	119
Tableau 5 – Attribut de variable normalisée à 2 octets	120
Tableau 6 – Codage de variable normalisée à 2 octets	120
Tableau 7 – Attribut de variable normalisée à 4 octets	121
Tableau 8 – Codage de variable normalisée à 4 octets	121
Tableau 9 – Attribut d'unipolaire à 2 octets	121
Tableau 10 – Codage d'unipolaire à 2 octets	121
Tableau 11 – Attribut d'une valeur virgule fixe à 2 Octets	122

Tableau 12 – Codage d'une valeur virgule fixe à 2 Octets	122
Tableau 13 – Attribut d'une valeur virgule fixe à 4 Octets.....	122
Tableau 14 – Codage de séquence de bits à 2 octets	123
Tableau 15 – Codage de quartet à 4 octets.....	123
Tableau 16 – Attribut de multiple de constante de temps de 2 octets	128
Tableau 17 – Attribut de multiple de constante de temps de 4 octets	129
Tableau 18 – Attribut de fraction de constante de temps de 2 octets.....	129
Tableau 19 – Codage de l'inverse de constante de temps de 2 octets.....	130
Tableau 20 – Base d'objets de gestion.....	136
Tableau 21 – Affectation de groupe d'accès.....	151
Tableau 22 – Attribution de droits d'accès.....	151
Tableau 23 – Services pour l'objet domaine	151
Tableau 24 – Service pour objet rapport	153
Tableau 25 – Services de l'entité gestion de FAL.....	161
Tableau 26 – Paramètre du service EM_DetectingDevice.....	162
Tableau 27 – Paramètre du service EM_OnlineReply.....	163
Tableau 28 – Paramètre du service EM_GetDeviceAttribute	164
Tableau 29 – Paramètre du service EM_ActiveNotification.....	166
Tableau 30 – Primitives du service EM_ConfiguringDevice.....	168
Tableau 31 – Paramètre du service EM_SetDefaultValue.....	170
Tableau 32 – Paramètres pour le service téléchargement aval dans un domaine	172
Tableau 33 – Paramètres pour le service téléchargement amont d'un domaine	174
Tableau 34 – Paramètres du service "EventReport"	175
Tableau 35 – Paramètres du service "AcknowledgeEventReport".....	176
Tableau 36 – Paramètres du service "ReportConditionChanging"	177
Tableau 37 – Paramètres du service "Read"	178
Tableau 38 – Paramètres du service "Write"	179
Tableau 39 – Paramètres du service "VariableDistribute".....	180
Tableau 40 – Paramètres du service FRTVariableDistribute.....	181
Tableau 41 – Paramètres du service FRTRead	182
Tableau 42 – Paramètres du service FRTWrite	183
Tableau 43 – Résumé des services de couche application	188

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 5-14: Définition des services de la couche application – Éléments de Type 14

AVANT-PROPOS

- 1) La CEI (Commission Électrotechnique Internationale) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de la CEI). La CEI a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, la CEI - entre autres activités - publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de la CEI"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec la CEI, participent également aux travaux. La CEI collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de la CEI concernant des questions techniques, représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de la CEI se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de la CEI. Tous les efforts raisonnables sont entrepris afin que la CEI s'assure de l'exactitude du contenu technique de ses publications; la CEI ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de la CEI s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de la CEI dans leurs publications nationales et régionales. Toute divergence entre toute Publication de la CEI et toute publication nationale ou régionale correspondante doit être indiquée en termes clairs dans cette dernière.
- 5) La CEI elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de la CEI. La CEI n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à la CEI, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de la CEI, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de la CEI ou de toute autre Publication de la CEI, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de la CEI peuvent faire l'objet de droits de propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

NOTE 1 L'utilisation de certains des types de protocoles associés est limitée par les détenteurs de leurs droits de propriété intellectuelle. Dans tous les cas, l'engagement à un abandon limité des droits de propriété intellectuelle pris par les détenteurs de ces droits permet d'utiliser un type particulier de protocole de couche liaison de données avec des protocoles de couche physique et de couche application dans des combinaisons de types telles que spécifiées de façon explicite dans les parties profil. L'utilisation des divers types de protocoles dans d'autres combinaisons peut exiger la permission donnée par les détenteurs respectifs de leurs droits de propriété intellectuelle.

La Norme internationale CEI 61158-5-14 a été établie par le sous-comité 65C: Réseaux de communication industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette seconde édition annule et remplace la première édition, parue en 2007. Cette édition constitue une révision technique.

Les principales modifications par rapport à l'édition précédente sont énumérées ci-dessous:

- Références normatives et Bibliographie mises à jour;
- Corrections d'erreurs éditoriales;
- Modifications de la spécification pour CPF3;
- Mise à jour des exigences pour toutes les classes de conformité;
- Mise à jour des exigences pour tous les services de conformité.

La présente version bilingue publiée en 2012-01 correspond à la version anglaise monolingue publiée en 2010-08.

Le texte anglais de cette norme est issu des documents 65C/606/FDIS et 65C/620/RVD.

Le rapport de vote 65C/620/RVD donne toute information sur le vote ayant abouti à l'approbation de cette norme.

La version française de cette norme n'a pas été soumise au vote.

Cette publication a été rédigée selon les Directives ISO/CEI, Partie 2.

Une liste de toutes les parties de la série CEI 61158, présentées sous le titre général *Réseaux de communication industriels – Spécifications de bus de terrain*, peut être consultée sur le site web de la CEI.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de la CEI sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. À cette date, la publication sera:

- reconduite;
- supprimée;
- remplacée par une édition révisée, ou
- amendée.

NOTE 2 La révision de la présente norme sera synchronisée avec les autres parties de la série CEI 61158.

INTRODUCTION

La présente partie de la CEI 61158 est l'une d'une série produite pour faciliter l'interconnexion de composants d'un système d'automation. Elle est apparentée à d'autres normes de la série telle que définie par le modèle de référence des bus de terrain "à trois couches" décrit dans le rapport IEC/TR 61158-1.

Le service application est fourni par le protocole d'application utilisant les services disponibles de la liaison de données ou autre couche immédiatement inférieure. La présente norme définit les caractéristiques de services d'application pouvant être exploitées par les applications de bus de terrain et/ou la gestion de système.

Dans toute la série de normes relatives aux bus de terrain, le terme "service" se réfère à la capacité abstraite fournie par une couche du Modèle de référence de base de l'interconnexion des systèmes ouverts (OSI) à la couche immédiatement supérieure. Ainsi, le service de la couche application défini dans la présente norme est un service architectural conceptuel, indépendant des divisions administrative et de mise en œuvre.

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 5-14: Définition des services de la couche application – Eléments de Type 14

1 Domaine d'application

1.1 Vue d'ensemble

La Couche application de bus de terrain (FAL¹) fournit aux programmes d'utilisateur un moyen d'accéder à l'environnement de communication du bus de terrain. À cet égard, la FAL peut être vue comme une «fenêtre entre des programmes d'application correspondants».

La présente norme fournit les éléments communs pour les communications de messagerie de base à temps critique et à temps non critique entre des programmes d'application dans un environnement d'automatisation et le matériel spécifique au bus de terrain de Type 14. Le terme "à temps critique" sert à représenter la présence d'une fenêtre temporelle, dans les limites de laquelle une ou plusieurs actions spécifiées sont tenues d'être parachevées avec un certain niveau défini de certitude. Le manquement à parachever les actions spécifiées dans les limites de la fenêtre temporelle risque d'entraîner la défaillance des applications qui demandent ces actions, avec le risque concomitant pour l'équipement, l'installation et éventuellement pour la vie humaine.

La présente norme définit de manière abstraite le service visible de l'extérieur fourni par la couche application de bus de terrain de Type 14 en termes

- a) d'un modèle abstrait pour définir des ressources (objets) d'application capables d'être manipulées par les utilisateurs par l'intermédiaire de l'utilisation du service FAL;
- b) des actions et événements primitifs du service;
- c) des paramètres associés à chaque action primitive et événement primitif, et la forme qu'ils prennent; et
- d) l'interrelation entre ces actions et événements, et leurs séquences valides.

Le but de la présente norme est de définir les services fournis à

- a) l'utilisateur de FAL à la frontière entre l'utilisateur et la couche application du modèle de référence de bus de terrain; et
- b) la Gestion des systèmes au niveau de la frontière entre la couche application et la Gestion des systèmes selon le modèle de référence de bus de terrain.

La présente norme spécifie la structure et les services de la couche application des bus de terrain de Type 14, en conformité avec le Modèle de référence de base de l'OSI (ISO/CEI 7498-1) et la structure de la couche application de l'OSI (ISO/CEI 9545).

Les services et protocoles de la FAL sont fournis par des entités d'application (AE²) de la FAL contenues dans les processus d'application. L'AE de la FAL se compose d'un jeu d'éléments de service application (ASE³) orientés objet et d'une entité de gestion de couche

1 FAL = *Fieldbus Application Layer*

2 AE = *Application Entity*

3 ASE = *Application Service Element*

(LME⁴) qui gère l'AE. Les ASE fournissent des services de communication qui fonctionnent sur un jeu de classes d'objets de processus application (APO⁵) connexes. L'un des ASE de la FAL est un ASE de gestion qui fournit un jeu commun de services pour la gestion des instances de classes de la FAL.

Bien que ces services spécifient, du point de vue des applications, la manière dont la demande et les réponses sont émises et délivrées, ils n'incluent pas une spécification de ce que les applications qui demandent et qui répondent doivent en faire. À savoir, les aspects comportementaux des applications ne sont pas spécifiés; seule une définition des demandes et réponses qu'elles peuvent envoyer/recevoir est spécifiée. Cela permet une plus grande flexibilité aux utilisateurs de la FAL pour normaliser un tel comportement d'objet. En plus de ces services, certains services d'appui sont également définis dans la présente norme pour fournir l'accès à la FAL afin de maîtriser certains aspects de son fonctionnement.

1.2 Spécifications

L'objectif principal de la présente norme est de spécifier les caractéristiques des services conceptuels d'une couche application qui sont adaptées à des communications à temps critique et, donc, complètent le Modèle de référence de base de l'OSI en guidant le développement des protocoles de couche application pour les communications à temps critique.

Un objectif secondaire est de fournir des trajets de migration à partir de protocoles industriels de communication préexistants. C'est ce dernier objectif qui donne naissance à la diversité des services normalisés tels que les divers types de la CEI 61158.

La présente spécification peut être utilisée comme la base pour les interfaces de programmation d'applications (Application Programming-Interfaces) formelles. Néanmoins, elle n'est pas une interface de programmation formelle et il sera nécessaire pour toute interface de ce type de traiter de questions de mise en œuvre qui ne sont pas couvertes par la présente spécification, y compris

- a) les tailles et l'ordonnancement des octets pour les divers paramètres de service à plusieurs octets, et
- b) la corrélation de primitives appariées "request-confirm" (demande et confirmation) ou "indication-response" (indication et réponse).

1.3 Conformité

La présente norme ne spécifie aucune mise en œuvre ou aucun produit individuels, de même qu'elle ne restreint nullement les mises en œuvre des entités de couche application dans les systèmes d'automation industriels.

Il n'y a pas de conformité d'équipement à la présente norme de définition des services de couche application. Au contraire, la conformité est obtenue par une mise en œuvre de protocoles conformes de couche application qui satisfont aux services de couche application de Type 14 définis dans la présente norme.

2 Références normatives

Les documents de référence suivants sont indispensables pour l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

⁴ LME = *Layer Management Entity*

⁵ APO = *Application Process Object*

IEC 61131-3, *Programmable controllers – Part 3: Programming languages* (disponible uniquement en anglais)

IEC 61158-4-14, *Industrial communication networks – Fieldbus specifications – Part 4-14: Data-link layer protocol specification – Type 14 elements* (disponible uniquement en anglais)⁶

IEC 61588, *Precision clock synchronization protocol for networked measurement and control systems* (disponible uniquement en anglais)

ISO/IEC 646, *Technologies de l'information – Jeu ISO de caractères codés à 7 éléments pour l'échange d'informations*

ISO/CEI 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

ISO/CEI 8822, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Définition du service de présentation*

ISO/CEI 8824-1, *Technologies de l'information – Notation de syntaxe abstraite numéro un (ASN.1) Spécification de la notation de base*

ISO/CEI 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche application*

ISO/CEI 10731, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Conventions pour la définition des services OSI*

3 Termes, définitions, symboles, abréviations et conventions

Pour les besoins du présent document, les termes suivants tels que définis dans ces publications s'appliquent:

3.1 Termes de l'ISO/CEI 7498-1

- a) entité d'application
- b) processus d'application
- c) unité de données de protocole application
- d) élément de service application
- e) invocation d'entité d'application
- f) invocation de processus d'application
- g) transaction d'application
- h) système ouvert réel
- i) syntaxe de transfert

3.2 Termes de l'ISO/CEI 8822

- a) syntaxe abstraite
- b) contexte de présentation

3.3 Termes de l'ISO/CEI 9545

- a) association d'applications (*application-association*)

⁶ Les publications monolingues des séries IEC 61158 et IEC 61784 sont actuellement en cours de traduction.

- b) contexte d'application (*application-context*)
- c) nom de contexte d'application (*application context name*)
- d) invocation d'entité d'application (*application-entity-invocation*)
- e) type d'entité d'application (*application-entity-type*)
- f) invocation de processus d'application (*application-process-invocation*)
- g) type de processus d'application (*application-process-type*)
- h) élément de service application (*application-service-element*)
- i) élément de service de contrôle d'application (*application control service element*)

3.4 Termes de l'ISO/CEI 8824-1

- a) identificateur d'objet
- b) type

3.5 Définitions spécifiques à la couche application des bus de terrain

3.5.1

contrôle d'accès

contrôle de la lecture et de l'écriture d'un objet

3.5.2

chemin d'accès

association d'un nom symbolique avec une variable dans le but d'une communication ouverte

3.5.3

macrocycle de communication

ensemble de cycles élémentaires nécessaires pour une activité de communication configurée dans un segment de macroréseau

3.5.4

programmation de communication

algorithmes et opération pour des transferts de données qui se produisent de manière déterministe et répétable

3.5.5

configuration (d'un système ou d'un dispositif)

étape dans la conception de système: action consistant à sélectionner des unités fonctionnelles, affecter leurs emplacements et définir leurs interconnexions

3.5.6

cyclique

répétitif d'une manière régulière

3.5.7

instance de FB de destination

instance de FB (bloc fonctionnel) qui reçoit les paramètres spécifiés

3.5.8

domaine

partie de mémoire utilisée pour stocker du code ou des données

3.5.9

téléchargement aval dans un domaine

opération d'écriture de données dans un domaine

3.5.10**téléchargement amont de domaine**

opération de lecture de données dans un domaine

3.5.11**entité**

chose particulière, telle qu'une personne, un lieu, un processus, un objet, un concept, une association ou un événement

3.5.12**pont**

entité relais de DL qui accomplit la synchronisation entre des liaisons (bus) et peut accomplir des fonctions sélectives de mise en mémoire et acheminement ultérieur et d'acheminement pour connecter deux segments de microréseau

3.5.13**identificateur**

mot de 16 bits associé à une variable du système

3.5.14**indice**

adresse d'un objet au sein d'un processus application

3.5.15**instance**

occurrence physique réelle d'un objet au sein d'une classe qui identifie l'un des plusieurs objets au sein de la même classe d'objets

3.5.16**instanciation**

création d'une instance d'un type spécifié

3.5.17**informations de gestion**

informations visibles du réseau servant à la gestion du système de terrain

3.5.18**base d'informations de gestion**

liste organisée d'informations de gestion

3.5.19**mise en correspondance**

ensemble de valeurs ayant une correspondance définie avec les grandeurs ou valeurs d'un autre ensemble

3.5.20**filtrage de message**

décision relative à un message conformément à une règle spéciale

3.5.21**microsegment**

partie intégrante d'un réseau où est mise en œuvre une programmation spéciale

3.5.22**décalage**

nombre d'octets à partir d'une position désignée spécialement

3.5.23

phase

fraction écoulee d'un cycle, mesurée à partir d'une certaine origine fixe

3.5.24

interface de processus

échange de données et mise en correspondance d'informations entre un processus physique et une unité d'application

3.5.25

temps réel

aptitude d'un système à fournir dans le délai imparti un résultat requis

3.5.26

communication temps réel

transfert de données en temps réel

3.5.27

Ethernet temps réel (RTE)

réseau selon l'ISO/CEI 8802-3 qui inclut la communication temps réel

NOTE 1 Une autre communication peut être prise en charge, à condition que la communication temps réel ne soit pas compromise.

NOTE 2 Cette définition est dédiée, mais sans que cela soit exhaustif, à l'ISO/CEI 8802-3. Elle pourrait s'appliquer à d'autres spécifications de l'IEEE 802, par exemple l'IEEE 802.11.

3.5.28

programme

configuration temporelle d'un certain nombre d'opérations connexes

3.5.29

macrocycle de programmation

intervalle de temps pour mettre en œuvre un programme spécifique

3.5.30

instance de FB source

instance de FB (bloc fonctionnel) qui envoie un paramètre spécifique

3.5.31

décalage temporel

différence temporelle par rapport à un temps spécialement désigné

3.6 Symboles et abréviations

AAE	Application Access Entity (Entité d'accès application)
AE	Application Entity (Entité d'application)
AL	Application Layer (Couche Application)
ALME	Application Layer Management Entity (Entité de gestion de couche application)
ALP	Application Layer Protocol (Protocole de couche application)
APO	Application Object (Objet d'application)
AP	Application Process (processus application)
APDU	Application Protocol Data Unit (Unité de donnée de protocole application)
API	Application Process Identifier (Identificateur de processus d'application)
AR	Application Relationship (Relation d'applications)

ARP	Address Resolution Protocol (protocole de résolution d'adresse)
AREP	Application Relationship End Point (Point d'extrémité de relation d'applications)
ASE	Application Service Element (Élément de service application)
Cnf	Confirmation
CR	Communication Relationship (Relation de communication)
CREP	Communication Relationship End Point (Point d'extrémité de relation de communication)
CSMA/CD	Carrier Sense Multiple Access Protocol with Collision Detection (Protocole d'accès multiple par surveillance du signal et détection de collision)
DD	Description de dispositif
DHCP	Dynamic Host Configuration Protocol (Protocole de configuration dynamique d'hôte)
DL-	(comme préfixe) Data Link (Liaison de données)
DLCEP	Data Link Connection End Point (Point d'extrémité de connexion de liaison de données)
DLL	Data Link Layer (Couche Liaison de données)
DLE	Data Link Entity (Entité de liaison de données)
DLM	Data Link-management (Gestion de liaison de données)
DLS	Data Link Service (Service de liaison de données)
DLSAP	Data link Service Access Point (Point d'accès au service liaison de données)
DLSDU	DL-Service-Data-Unit (Unité de données de service de DL)
ECSME	Type 14 communication scheduling management entity (Entité de gestion de programmation pour communication de Type 14)
Type 14	Ethernet pour automation de centrale
EM_	(comme préfixe) Gestion de Type 14
ESME	Type 14 Socket Mapping Entity (Entité de mapping de ports de type 14)
FB	Function Block (Bloc fonctionnel)
FBAP	Function Block Application Process (Processus application de bloc fonctionnel)
FRT	Fast Real-time (Temps réel rapide)
Ind	Indication
IP	Internet Protocol (Protocole Internet)
LLC	Logical Link Control (Commande de liaison logique)
LMP	Link Management Protocol (Protocole de gestion de liaison)
MAC	Medium Access Control (Commande d'accès au support)
MAU	Medium Attachment Unit (Unité de raccordement au support)
MOB	Management Object Base (Bases d'objets de gestion)
PAD	Pad (bits de bourrage)
PDU	Protocol Data Unit (Unité de données de protocole)
P/S	Publisher/Subscriber (Éditeur/Abonné)
Req	Request (Demande)
Rsp	Response (Réponse)
RTE	Real-Time Ethernet (Ethernet temps réel)
RT-Ethernet	Real-Time Ethernet (Ethernet temps réel)
SAP	Service Access Point (Point d'accès au service)
SDU	Service Data Unit (Unité de donnée de service)

SME	System Management Entity (Entité de gestion de système)
SNTP	Simple Network Time Protocol (Protocole de temps réseau simple)
TCP	Transmission Control Protocol (Protocole de commande de transmission)
UDP	User Datagram Protocol (Protocole datagramme d'utilisateur)

3.7 Conventions

3.7.1 Vue d'ensemble

La couche FAL est définie comme un jeu d'éléments ASE orientés objet. Chaque ASE est spécifié dans un paragraphe distinct. Chaque spécification d'ASE est constituée de deux parties, à savoir sa spécification de classe et sa spécification de service.

La spécification de classe définit les attributs de la classe. Les attributs sont accessibles à partir d'instances de la classe en utilisant les services d'ASE de gestion d'objets spécifiés à l'Article 5. La spécification de services définit les services qui sont fournis par l'ASE.

3.7.2 Conventions pour les définitions de classes

Les définitions de classes sont décrites à l'aide de modèles. Chaque modèle est constitué d'une liste d'attributs de la classe. La forme générale du modèle est montrée ci-dessous:

FAL ASE:	Nom de l'ASE
CLASS:	Nom de la classe
CLASS ID:	#
PARENT CLASS:	Nom de la classe parent
ATTRIBUTES:	
1 (o) Attribut clé:	Identificateur numérique
2 (o) Attribut clé:	nom
3 (m) Attribut:	nom d'attribut(valeurs)
4 (m) Attribut:	nom d'attribut(valeurs)
4.1 (s) Attribut:	nom d'attribut(valeurs)
4.2 (s) Attribut:	nom d'attribut(valeurs)
4.3 (s) Attribut:	nom d'attribut(valeurs)
5. (c) Contrainte:	expression de la contrainte
5.1 (m) Attribut:	nom d'attribut(valeurs)
5.2 (o) Attribut:	nom d'attribut(valeurs)
6 (m) Attribut:	nom d'attribut(valeurs)
6.1 (s) Attribut:	nom d'attribut(valeurs)
6.2 (s) Attribut:	nom d'attribut(valeurs)
SERVICES:	
1 (o) OpsService:	nom du service
2. (c) Contrainte:	expression de la contrainte
2.1 (o) OpsService:	nom du service
3 (m) MgtService:	nom du service

- (a) La rubrique "FAL ASE:" est le nom de l'élément ASE de la couche FAL (FAL ASE) qui fournit les services pour la classe spécifiée.
- (b) La rubrique "CLASS:" est le nom de la classe spécifiée. Tous les objets définis à l'aide de ce modèle seront une instance de cette classe. La classe peut être spécifiée par la présente norme ou par un utilisateur de la présente norme.
- (c) La rubrique "CLASS ID:" est un numéro qui identifie la classe spécifiée. Ce numéro est unique au sein du FAL ASE qui fournira les services pour cette classe. Lorsqu'il est qualifié par l'identité de son FAL ASE, il identifie sans ambiguïté la classe relevant du domaine d'application de la FAL. La valeur "NULL" indique que la classe ne peut pas être

instanciée. Les Class ID (identificateurs de classe) entre 1 et 255 sont réservés par la présente norme pour identifier des classes normalisées. Ils ont été attribués pour maintenir la compatibilité avec des normes nationales existantes. Les CLASS ID entre 256 et 2048 sont alloués pour identifier les classes définies par l'utilisateur.

- (c) La rubrique "PARENT CLASS:" est le nom de la classe parent pour la classe spécifiée. Tous les attributs définis pour la classe parent et hérités par celle-ci sont hérités pour la classe définie, et ils n'ont donc pas à être redéfinis dans le modèle pour cette classe.

NOTE La classe parent "TOP" indique que la classe définie est une définition de classe initiale. La classe parent "TOP" est utilisée comme point de départ à partir duquel toutes les autres classes sont définies. L'usage de "TOP" est réservé pour les classes définies par la présente norme.

- (d) L'étiquette "ATTRIBUTES" (Attributs) indique que les entrées suivantes sont des attributs définis pour la classe.

- 1) Chacune des entrées d'attribut contient un numéro de ligne dans la colonne 1, un indicateur obligatoire (m) / facultatif (o) / conditionnel (c) / sélecteur (s) dans la colonne 2, une étiquette de type d'attribut dans la colonne 3, un nom ou une expression conditionnelle dans la colonne 4, et, facultativement, une liste de valeurs énumérées dans la colonne 5. Dans la colonne suivant la liste de valeurs, la valeur par défaut pour l'attribut peut être spécifiée.
- 2) Les objets sont normalement identifiés par un identificateur numérique et/ou par un nom d'objet. Dans les modèles de classe, ces attributs clés sont définis sous l'attribut clé.
- 3) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point. L'imbrication est utilisée pour spécifier
 - i) des champs d'un attribut structuré (4.1, 4.2, 4.3),
 - ii) des attributs conditionnés à un énoncé de contrainte (5). Les attributs peuvent être obligatoires (5.1) ou facultatifs (5.2) si la contrainte est vraie. Tous les attributs facultatifs n'exigent pas des énoncés de contraintes comme le fait l'attribut défini en (5.2).
 - iii) les champs sélection d'un attribut de type choix (6.1 et 6.2).

- (e) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe.

- 1) Un (m) dans la colonne 2 indique que le service est obligatoire pour la classe, alors qu'un (o) indique qu'il est facultatif. Un (c) dans cette colonne indique que le service est conditionnel. Lorsque tous les services définis pour une classe le sont comme étant facultatifs, l'un au moins doit être sélectionné quand une instance de la classe est définie.
- 2) L'étiquette "OpsService" désigne un service opérationnel (1).
- 3) L'étiquette "MgtService" désigne un service de gestion (2).
- 4) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point. L'imbrication dans la liste de services sert à spécifier des services conditionnés à un énoncé de contrainte.

3.7.3 Conventions pour les définitions des services

3.7.3.1 Généralités

La présente norme utilise les conventions descriptives données dans l'ISO/CEI 10731.

Le modèle de service, les primitives de service et les diagrammes de temps-séquence utilisés sont des descriptions totalement abstraites; ils ne constituent pas une spécification pour une mise en œuvre.

3.7.3.2 Paramètres de service

Les primitives de service sont utilisées pour représenter les interactions entre utilisateur de service et fournisseur de service (ISO/CEI 10731). Elles acheminent des paramètres qui indiquent des informations disponibles dans l'interaction entre utilisateur et fournisseur.

NOTE 1 Voir la NOTE en 3.7.3.3 relative à la non-inclusion des paramètres de service qui sont appropriés à la spécification de protocole ou à la spécification d'interface de programmation ou à la spécification de mise en œuvre, mais pas à une définition de service abstrait.

La présente norme utilise un format de tableau pour décrire les paramètres de composants des primitives de service. Les paramètres qui s'appliquent à chaque groupe de primitives de service sont consignés en tableaux dans toute la suite de la présente norme. Chaque tableau comporte jusqu'à six colonnes: une colonne pour le nom du paramètre de service et une colonne chacune pour les primitives et les sens de transfert de paramètres utilisés par le service. Les six colonnes possibles sont:

- a) le nom de paramètre;
- b) les paramètres d'entrée de la primitive "request";
- c) les paramètres de sortie de la primitive "request";

NOTE 2 Il s'agit d'une fonctionnalité rarement utilisée. Sauf spécification contraire, les paramètres de primitive "request" sont des paramètres d'entrée.

- d) les paramètres de sortie de la primitive "indication";
- e) les paramètres d'entrée de la primitive "response", et
- f) les paramètres de sortie de la primitive "confirm".

NOTE 3 Les primitives "request", "indication", "response" et "confirm" sont aussi respectivement appelées primitives "requestor.submit", "acceptor.deliver", "acceptor.submit", et "requestor.deliver" (voir ISO/CEI 10731).

Les spécifications de service selon la présente norme utilisent un format de tableau pour décrire les paramètres de composants des primitives de service d'ASE. Les paramètres qui s'appliquent à chaque groupe de primitives de service sont consignés en tableaux. Chaque tableau comporte jusqu'à cinq colonnes pour le/la:

- a) nom de paramètre,
- b) primitive "request",
- c) primitive "indication",
- d) primitive "response", et
- e) primitive "confirm".

Un paramètre (ou un composant de celui-ci) est énuméré dans chaque rangée de chaque tableau. Dans les colonnes appropriées de la primitive de service, un code est utilisé pour spécifier le type d'usage du paramètre sur la primitive spécifiée dans la colonne:

- M Le paramètre est obligatoire pour la primitive;
- U Le paramètre est une option de l'utilisateur et peut ou peut ne pas être fourni, cela dépendant de l'usage dynamique de l'utilisateur du service. Lorsqu'il n'est pas fourni, une valeur par défaut est supposée pour le paramètre;
- C Le paramètre est conditionné à d'autres paramètres ou à l'environnement de l'utilisateur du service;
- (blanc/vidé) le paramètre n'est jamais présent.
- S Le paramètre est un élément sélectionné.

Certaines entrées sont en plus qualifiées par des éléments entre parenthèses. Ceux-ci peuvent être

- une contrainte spécifique au paramètre:

"(=)" indique que le paramètre équivaut du point de vue de la sémantique au paramètre dans la primitive de service située immédiatement à sa gauche dans le tableau;

- une indication qu'une certaine note s'applique à l'entrée:

"(n)" indique que la note "n" suivante contient des informations complémentaires relatives au paramètre et à son utilisation.

3.7.3.3 Procédures de service

Les procédures sont définies en termes des

- interactions entre entités d'application par l'échange d'unités de données de protocole d'application de bus de terrain, et
- interactions entre un fournisseur de services de couche application et un utilisateur de service de couche application dans le même système par l'invocation de primitives de service de couche application.

Ces procédures sont applicables à des instances de communication entre systèmes qui prennent en charge des services de communications à contrainte temporelle au sein de la couche application de bus de terrain.

NOTE La sous-série de normes CEI 61158-5 définit les jeux de services abstraits. Il ne s'agit ni de spécifications de protocole ni de spécifications de mise en œuvre ni de spécifications d'interface de programmation concrète. Par conséquent, il y a des restrictions concernant la mesure dans laquelle les procédures de service peuvent être mandatées dans les parties de la sous-série CEI 61158-5. Les aspects de protocole qui peuvent varier parmi différentes spécifications de protocole ou différentes mises en œuvre quiinstancient les mêmes services abstraits ne sont pas appropriés à être inclus dans ces définitions de service, excepté au niveau d'abstraction qui est nécessairement commun à toutes ces expressions.

Par exemple, le moyen par lequel des fournisseurs de service appariaient des PDU de demande et de réponse est approprié pour la spécification dans une norme de spécification de protocole de la CEI 61158-6 mais pas dans une norme de définition de services abstraits de la sous-série CEI 61158-5. De même, les méthodes de mise en œuvre locales par lesquelles un fournisseur de service ou utilisateur de service appariaient des primitives "request" et "confirm", ou des primitives "indication" et "response", sont appropriées pour une spécification de mise en œuvre ou pour une spécification d'interface de programmation, mais pas pour une norme de service abstrait ou pour une norme de protocole, excepté à un niveau d'abstraction qui est nécessairement commun à tous les modes de réalisation de la norme de spécification. Dans tous les cas, la définition abstraite n'est pas autorisée à spécifier outre mesure la réalisation d'instanciation plus concrète.

Des informations supplémentaires sur les procédures de service conceptuelles d'une mise en œuvre d'un protocole qui réalise les services de l'une des définitions de service abstrait de la sous-série CEI 61158-5 peuvent être consultées dans le rapport CEI/TR 61158-1:2007, 9.6.

4 Concepts

Les concepts et modèles communs utilisés pour décrire le service de couche application dans la présente norme sont détaillés dans le rapport CEI/TR 61158-1, Article 9.

5 Data type ASE

5.1 Vue d'ensemble

Une vue d'ensemble du type de données ASE et des relations entre types de données est fournie dans le rapport CEI/TR 61158-1, 10.1.

5.2 Définition formelle des objets de types de données

Le modèle utilisé pour décrire la classe des types de données dans le présent article est détaillé dans le rapport CEI/TR 61158-1, 10.2. Ceci inclut la structure spécifique de l'ASE et la définition des ses attributs.

5.3 Types de données définis pour la FAL

5.3.1 Types Fixed length (longueur fixe)

5.3.1.1 Types Boolean (booléens)

5.3.1.1.1 Boolean

CLASS: Data type (Type de données)

ATTRIBUTS:

1	Data type numeric identifier	=	1
2	Data type Name	=	Boolean
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

Ce type de données exprime le type de données Boolean avec les valeurs TRUE et FALSE.

5.3.1.1.2 BOOL

Ce type de la CEI 61131-3 est le même que le type Boolean.

5.3.1.1.3 VT_BOOLEAN

CLASS: Type de données

ATTRIBUTS:

2	Data type Name =	VT_BOOLEAN
4	Format	= FIXED LENGTH
4.1	Octet Length	= 2

Ce type de données exprime un type de données Boolean avec les valeurs TRUE (-1) et FALSE (0) (voir Interger16).

5.3.1.2 Types Bitstring (chaîne de bits)

5.3.1.2.1 BitString8

CLASS: Type de données

ATTRIBUTS:

1	Data type Numeric Identifier	=	22
2	Data type Name	=	Bitstring8
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	1

Ce type contient un élément de type BitString.

5.3.1.2.2 OCTET

Ce type de la CEI 61131-3 est le même que le type Bitstring8.

5.3.1.2.3 BitString16

CLASS: Type de données

ATTRIBUTS:

1	Data type Numeric Identifier	=	23
2	Data type Name	=	Bitstring16
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	2

5.3.1.2.4 WORD

Ce type de la CEI 61131-3 est le même que le type Bitstring16.

5.3.1.2.5 BitString32**CLASS:** Type de données**ATTRIBUTES:**

1	Data type Numeric Identifier	=	24
2	Data type Name	=	Bitstring32
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	4

5.3.1.2.6 DWORD

Ce type de la CEI 61131-3 est le même que le type Bitstring32.

5.3.1.2.7 BitString64**CLASS:** Type de données**ATTRIBUTES:**

1	Data type Numeric Identifier	=	57
2	Data type Name	=	Bitstring64
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	8

5.3.1.2.8 LWORD

Ce type de la CEI 61131-3 est le même que le type Bitstring64.

5.3.1.3 Types de données**5.3.1.3.1 BinaryDate****CLASS:** Type de données**ATTRIBUTES:**

1	Data type Numeric Identifier	=	11
2	Data type Name	=	BinaryDate
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

Ce type de données est constitué de six éléments de valeurs unsigned (non signées) et exprime l'heure et la date calendaires. Le premier élément est un type de données Unsigned16 et donne la fraction d'une minute en millisecondes. Le deuxième élément est un type de données Unsigned8 et donne la fraction d'une heure en minutes. Le troisième élément est un type de données Unsigned8 et donne la fraction d'un jour en heures. Le quatrième élément est un type de données Unsigned8. Ses trois (3) bits de poids fort donnent le jour de la semaine et ses cinq (5) bits de poids faible donnent le jour du mois. Le cinquième élément est un type de données Unsigned8 et donne le mois. Le dernier élément est un type de données Unsigned8 et donne l'année.

5.3.1.3.2 BinaryDate2000**CLASS:** Type de données**ATTRIBUTES:**

1	Data type Numeric Identifier	=	51
2	Data type Name	=	BinaryDate2000
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

Ce type de données est constitué de six éléments de valeurs unsigned (non signées) et exprime l'heure et la date calendaires. Le premier élément est un type de données Unsigned16 et donne la fraction d'une minute en millisecondes. Le deuxième élément est un type de données Unsigned8 et donne la fraction d'une heure en minutes. Le troisième

élément est un type de données Unsigned8 et donne la fraction d'un jour en heures. Le quatrième élément est un type de données Unsigned8. Ses trois (3) bits de poids fort donnent le jour de la semaine et ses cinq (5) bits de poids faible donnent le jour du mois. Le cinquième élément est un type de données Unsigned8 et donne le mois. Le dernier élément est un type de données Unsigned16 et donne l'année.

5.3.1.3.3 Date

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	50
2	Data type Name	=	Date
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	7

Ce type de données est constitué de six éléments de valeurs unsigned (non signées) et exprime l'heure et la date calendaires. Le premier élément est un type de données Unsigned16 et donne la fraction d'une minute en millisecondes. Le deuxième élément est un type de données Unsigned8 et donne la fraction d'une heure en minutes. Le troisième élément est un type de données Unsigned8 et donne la fraction d'un jour en heures, le bit de poids fort indiquant l'heure normalisée (Standard Time) ou l'heure d'économie d'énergie (Daylight Saving Time). Le quatrième élément est un type de données Unsigned8. Ses trois (3) bits de poids fort donnent le jour de la semaine et ses cinq (5) bits de poids faible donnent le jour du mois. Le cinquième élément est un type de données Unsigned8 et donne le mois. Le dernier élément est un type de données Unsigned8 et donne l'année. Les valeurs 0 ... 50 correspondent aux années 2000 à 2050, les valeurs 51 ... 99 correspondent aux années 1951 à 1999.

5.3.1.3.4 DATE

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	non utilisé
2	Data type Name	=	DATE
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

Ce type de la CEI 61131-3 est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de deux octets. Il exprime la date comme un nombre de jours, commençant à partir de 1972.01.01 (1^{er} janvier 1972), début de l'ère du temps universel coordonné (TUC), jusqu'au 2151.06.06 (6 juin 2151), c'est-à-dire une plage totale de 65 536 jours.

5.3.1.3.5 Date

Ce type de données est le même que le type Float64.

Le type de données date a une résolution de l'ordre d'une nanoseconde. Il est valide pour les dates entre le 1^{er} janvier 0100 et le 31 décembre 9999. La valeur 0,0 a été définie pour le 30 décembre 1899, 00:00. La partie entière de la valeur représente les jours après le 30 décembre 1899 (pour les dates antérieures à ce jour, la valeur correspondante est négative); la partie fractionnaire définit l'heure du jour en question.

5.3.1.3.6 TimeOfDay**CLASS:** Type de données**ATTRIBUTS:**

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 12 |
| 2 | Data type Name | = | TimeOfDay |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 6 |

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) et exprime l'heure du jour et la date. Le premier élément est un type de données Unsigned32 et donne l'heure après minuit en millisecondes. Le second élément est un type de données Unsigned16 et donne la date en comptant les jours à partir du 1^{er} janvier 1984.

5.3.1.3.7 TimeOfDay with date indication (avec indication de date)

Ce type de données est le même que le type de données TimeOfDay défini ci-dessus.

5.3.1.3.8 TimeOfDay without date indication (sans indication de date)**CLASS:** Type de données**ATTRIBUTS:**

- | | | | |
|-----|------------------------------|---|---|
| 1 | Data type Numeric Identifier | = | 52 |
| 2 | Data type Name | = | TimeOfDay without date indication (sans indication de date) |
| 4 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

Ce type de données est constitué d'un seul élément d'une valeur non signée et exprime l'heure du jour. L'élément est un type de données Unsigned32 et donne l'heure après minuit en millisecondes.

5.3.1.3.9 TIME_OF_DAY

Ce type de la CEI 61131-3 est le même que le type TimeOfDay without date indication.

5.3.1.3.10 PrecisionTimeDifference**CLASS:** Type de données**ATTRIBUTS:**

- | | | | |
|-----|------------------------------|---|-------------------------|
| 1 | Data type Numeric Identifier | = | 13 |
| 2 | Data type Name | = | PrecisionTimeDifference |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 ou 8 |

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) et exprime la différence de temps. Le premier élément facultatif est un type de données Unsigned32 qui fournit la différence en secondes. Le dernier élément est un type de données signed32 qui donne une partie fractionnaire d'une seconde en microsecondes.

5.3.1.3.11 PrecisionTimeDifference with second indication (avec indication de secondes)**CLASS:** Type de données**ATTRIBUTS:**

- | | | | |
|-----|------------------------------|---|--|
| 1 | Data type Numeric Identifier | = | 53 |
| 2 | Data type Name | = | PrecisionTimeDifference with second indication (avec une indication de secondes) |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 8 |

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) et exprime la différence temporelle. Le premier élément est un type de données Unsigned32 qui fournit la différence en secondes. Le dernier élément est un type de données signed32 qui donne une partie fractionnaire d'une seconde en microsecondes.

5.3.1.3.12 PrecisionTimeDifference without second indication (sans indication de secondes)

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|---|
| 1 | Data type Numeric Identifier | = | 54 |
| 2 | Data type Name | = | PrecisionTimeDifference without second indication (sans indication de secondes) |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

Ce type de données est une valeur signée qui exprime la différence temporelle en microsecondes.

5.3.1.3.13 TimeValue

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 21 |
| 2 | Data type Name | = | Time Value |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 8 |

Ce type simple exprime l'heure ou la différence temporelle en un nombre binaire en complément à deux avec une longueur de huit octets. L'unité de temps est 1/32 milliseconde.

5.3.1.3.14 UniversalTime

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|---------------|
| 1 | Data type Numeric Identifier | = | 16 |
| 2 | Data type Name | = | UniversalTime |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 12 |

Ce type simple est constitué de douze éléments de type VisibleString. (AAMMJJHHMMSS). Il est le même que celui défini dans l'ISO/CEI 8824-1, excepté que le différentiel de temps local n'est pas pris en charge.

5.3.1.3.15 FieldbusTime

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 17 |
| 2 | Data type Name | = | FieldbusTime |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 7 |

Ce type de données est défini dans la CEI 61158-4-14 de la présente norme comme DL-Time (temps DL).

5.3.1.4 Types numériques

5.3.1.4.1 BCD

CLASS: Type de données

ATTRIBUTS:

1	Data type Numeric Identifier	=	35
2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur égale à un octet. Dans ce type, les quatre bits de poids faible sont utilisés pour exprimer une valeur BCD (ou DCB «décimal codé binaire») qui est comprise entre zéro et neuf inclus. Les quatre bits de poids fort ne sont pas utilisés.

5.3.1.4.2 Types Floating Point (virgule flottante)

5.3.1.4.2.1 Float32

CLASS: Type de données

ATTRIBUTS:

1	Data type Numeric Identifier	=	8
2	Data type Name	=	Float32
4	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type a une longueur de quatre octets. Le format pour float32 est celui défini par l'ANSI/IEEE 754 comme single precision ("simple précision").

5.3.1.4.2.2 Floating-point (virgule flottante)

Ce type de données est le même que le type Float32.

5.3.1.4.2.3 REAL

Ce type de la CEI 61131-3 est le même que le type Float32.

5.3.1.4.2.4 float

Ce type de données est le même que le type Float32.

5.3.1.4.2.5 Float64

CLASS: Type de données

ATTRIBUTS:

1	Data type Numeric Identifier	=	15
2	Data type Name	=	Float64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

Ce type a une longueur de huit octets. Le format pour Float64 est celui défini par l'ANSI/IEEE 754 comme double precision ("double précision").

5.3.1.4.2.6 LREAL

Ce type de la CEI 61131-3 est le même que le type Float64.

5.3.1.4.2.7 double

Ce type de données est le même que le type Float64.

5.3.1.4.3 Types Integer (entiers)

5.3.1.4.3.1 Integer8

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 2 |
| 2 | Data type Name | = | Integer8 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 1 |

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à un octet.

5.3.1.4.3.2 SINT

Ce type de la CEI 61131-3 est le même que le type Integer8.

5.3.1.4.3.3 char

Ce type de données est le même que le type Integer8.

5.3.1.4.3.4 Integer16

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 3 |
| 2 | Data type Name | = | Integer16 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 2 |

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à deux octets.

5.3.1.4.3.5 INT

Ce type de la CEI 61131-3 est le même que le type Integer16.

5.3.1.4.3.6 Short

Ce type de données est le même que le type Integer16.

5.3.1.4.3.7 Integer32

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 4 |
| 2 | Data type Name | = | Integer32 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à quatre octets.

5.3.1.4.3.8 DINT

Ce type de la CEI 61131-3 est le même que le type Integer32.

5.3.1.4.3.9 Long

Ce type de données est le même que le type Integer32.

5.3.1.4.3.10 Integer64

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	55
2	Data type Name	=	Integer64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

Ce type entier (Integer) est un nombre binaire en complément à deux avec une longueur égale à huit octets.

5.3.1.4.3.11 LINT

Ce type de la CEI 61131-3 est le même que le type Integer64.

5.3.1.4.4 Types Unsigned (non signés)**5.3.1.4.4.1 Unsigned8**

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	5
2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur égale à un octet.

5.3.1.4.4.2 USINT

Ce type de la CEI 61131-3 est le même que le type Unsigned8.

5.3.1.4.4.3 unsigned char

Ce type de données est le même que le type Unsigned8.

5.3.1.4.4.4 Unsigned16

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	6
2	Data type Name	=	Unsigned16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de deux octets.

5.3.1.4.4.5 UINT

Ce type de la CEI 61131-3 est le même que le type Unsigned16.

5.3.1.4.4.6 unsigned short

Ce type de données est le même que le type Unsigned16.

5.3.1.4.4.7 Unsigned32

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	7
2	Data type Name	=	Unsigned32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de quatre octets.

5.3.1.4.4.8 UDINT

Ce type de la CEI 61131-3 est le même que le type Unsigned32.

5.3.1.4.4.9 unsigned long

Ce type de données est le même que le type Unsigned32.

5.3.1.4.4.10 Unsigned64

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	56
2	Data type Name	=	Unsigned64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type unsigned (non signé) a une longueur de huit octets.

5.3.1.4.4.11 ULINT

Ce type de la CEI 61131-3 est le même que le type Unsigned64.

5.3.1.4.4.12 Variable normalisée à 2 octets

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	113
2	Data type Name	=	N2
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

Valeur normalisée linéaire. 0 % correspond à 0 (0x0), 100 % correspond à 214 (0x4000).

La représentation en complément à deux, le MSB («Most Significant Bit» bit de poids fort) est le bit après le bit de signe (SN) du premier octet. L'attribut de type de données et le poids de chaque bit sont montrés dans le Tableau 1 et dans le Tableau 2.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs.

Tableau 1 – Attribut de variable normalisée à 2 octets

Type de données	Plage de valeurs	Résolution	Longueur
N2	$-200 \% \leq i \leq (200 \cdot 2^{-14}) \%$	$2^{-14} = 0,006 \text{ 1 } \%$	2 Octets

Tableau 2 – Codage de variable normalisée à 2 octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}

5.3.1.4.4.13 Variable normalisée à 4 octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 114
- 2 Data type Name = N4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Valeur normalisée linéaire. 0 % correspond à 0 (0x0), 100 % correspond à 230 (0x4000 000).

La représentation en complément à deux, le MSB («Most Significant Bit» bit de poids fort) est le bit après le bit de signe (SN) du premier octet. L'attribut de type de données et le poids de chaque bit sont montrés dans le Tableau 3 et dans le Tableau 4.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs.

Tableau 3 – Attribut de variable normalisée à 4 octets

Type de données	Plage de valeurs	Résolution	Longueur
N4	$-200 \% \leq i \leq (200 \cdot 2^{-30}) \%$	$2^{-30} = 9,3 \cdot 10^{-8} \%$	4 Octets

Tableau 4 – Codage de variable normalisée à 4 octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2				:				
3				:				
4	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}

5.3.1.4.4.14 Variable normalisée à 2 octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 123
- 2 Data type Name = X2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Valeur normalisée linéaire. 0 % correspond à 0 (0x0), 100 % correspond à 2X. La structure est la même que dans le cas des types de données N2 et N4, mais la normalisation (100 %) ne renvoie automatiquement pas respectivement au bit 14 ou au bit 30, mais est variable. Le bit de normalisation est codé en un paramètre complémentaire.

La représentation en complément à deux, le MSB («Most Significant Bit» bit de poids fort) est le bit après le bit de signe (SN) du premier octet. L'attribut de type de données et le poids de chaque bit sont montrés dans le Tableau 5 et dans le Tableau 6.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs.

Tableau 5 – Attribut de variable normalisée à 2 octets

Type de données	Plage de valeurs	Résolution	Longueur
X2 (par exemple avec x = 12)	$-800 \% \leq i \leq (800 \cdot 2^{-14}) \%$	2^{-12}	2 Octets

Tableau 6 – Codage de variable normalisée à 2 octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}

5.3.1.4.4.15 Variable normalisée à 4 octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 124
- 2 Data type Name = X4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Valeur normalisée linéaire. 0 % correspond à 0 (0x0), 100 % correspond à 2X. La structure est la même que dans le cas des types de données N2 et N4, mais la normalisation (100 %) ne renvoie automatiquement pas respectivement au bit 14 ou au bit 30, mais est variable. Le bit de normalisation est codé en un paramètre complémentaire.

La représentation en complément à deux, le MSB («Most Significant Bit» bit de poids fort) est le bit après le bit de signe (SN) du premier octet. L'attribut de type de données et le poids de chaque bit sont montrés dans le Tableau 7 et dans le Tableau 8.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs.

Tableau 7 – Attribut de variable normalisée à 4 octets

Type de données	Plage de valeurs	Résolution	Longueur
X4 (par exemple avec x = 28)	$-800 \% \leq i \leq (800 \cdot 2^{-30}) \%$	2^{-28}	2 Octets

Tableau 8 – Codage de variable normalisée à 4 octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
2				:				
3				:				
4	2^{-23}	2^{-24}	2^{-25}	2^{-26}	2^{-27}	2^{-28}	2^{-29}	2^{-30}

5.3.1.4.4.16 Unipolaire à 2 octets

CLASS: Type de données

ATTRIBUTS:

1 Data type Numeric Identifier = ~~125~~

2 Data type Name = Unipolar 2.16

3 Format = ~~FIXED LENGTH~~

4.1 Octet Length = 2

Un type de primitive avec des valeurs distinguées qui sont non négatives, des nombres entiers divisés par une puissance fixe de deux, exprimant la valeur en pourcentage d'une étendue.

NOTE 1 Ces types n'existent pas dans l'ASN 1, ils sont exprimés dans la CEI 870 comme "nombre virgule fixe non signé"

NOTE 2 Ces types sont définis dans la CEI 61375

Le nombre avant la virgule donne le nombre puissance de 2 formant la partie entière. Le facteur epsilon est égal à la valeur de la plus petite puissance de deux dans le mot. Il convient qu'une variable de type unipolaire soit émise comme entier non signé. L'attribut de type de données et le poids de chaque bit sont montrés dans le Tableau 9 et dans le Tableau 10.

Tableau 9 – Attribut d'unipolaire à 2 octets

Type de données	Plage de valeurs	Résolution	Longueur
Unipolar2.16	$0 \% \leq i \leq (400 \cdot 2^{-14}) \%$	2^{-14}	2 Octets

Tableau 10 – Codage d'unipolaire à 2 octets

[illegible]

5.3.1.4.4.17 Valeur virgule fixe à 2 octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 121
- 2 Data type Name = E2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Valeur virgule fixe linéaire avec sept binaires après la virgule décimale. 0 correspond à 0 (0x0), 128 correspond à 214 (0x4000).

La représentation en complément à deux, le MSB («Most Significant Bit» bit de poids fort) est le bit après le bit de signe (SN) du premier octet. L'attribut de type de données et le poids de chaque bit sont montrés dans le Tableau 11 et dans le Tableau 12.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs.

Tableau 11 – Attribut d'une valeur virgule fixe à 2 Octets

Type de données	Plage de valeurs	Résolution	Longueur
E2	$-256+2^{-7} \leq i \leq 256-2^{-7}$	$2^{-7} = 0,007\ 812\ 5$	2 Octets

Tableau 12 – Codage d'une valeur virgule fixe à 2 Octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
2	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

5.3.1.4.4.18 Valeur virgule fixe à 4 octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 122
- 2 Data type Name = C4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Valeur virgule fixe linéaire à quatre décimales. 0 correspond à 0 (0x0), 0,000 1 correspond à 2^0 (0x0000 0001).

Comme dans le cas de Integer32, la pondération des bits a été réduite d'un facteur 10 000. L'attribut de type de donnée est montré dans le Tableau 13.

Tableau 13 – Attribut d'une valeur virgule fixe à 4 Octets

Type de données	Plage de valeurs	Résolution	Longueur
C4	$-214\ 748,364\ 8 \leq i \leq 214\ 748,364\ 7$	$10^{-4} = 0.0001$	4 Octets

5.3.1.4.4.19 Séquence de bits à 2 Octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 115
- 2 Data type Name = V2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Séquence de bits pour contrôler et représenter des fonctions d'application. 16 variables Boolean sont combinées en deux octets. Le poids de chaque bit est montré dans le Tableau 14.

Tableau 14 – Codage de séquence de bits à 2 octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	15	14	13	12	11	10	9	8
2	7	6	5	4	3	2	1	0

5.3.1.4.4.20 Quartet à 4 Octets

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 116
- 2 Data type Name = L2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Quatre bits associés forment un quartet (ou demi-octet). Quatre quartets sont représentés en deux octets.

La définition du quartet n'est pas spécifiée. Le poids de chaque bit est montré dans le Tableau 15.

Tableau 15 – Codage de quartet à 4 octets

Octet	Bit							
	7	6	5	4	3	2	1	0
1	Quartet 3				Quartet 2			
2	Quartet 1				Quartet 0			

5.3.1.5 Types de caractères OctetString

5.3.1.5.1 OctetString1

CLASS: Type de données

ATTRIBUTS:

- 1 Data type Numeric Identifier = 30
- 2 Data type Name = OctetString1
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 1

Ce type a une longueur égale à un octet.

5.3.1.5.2 OctetString2

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 31 |
| 2 | Data type Name | = | OctetString2 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 2 |

Ce type a une longueur de deux octets.

5.3.1.5.3 OctetString4

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 32 |
| 2 | Data type Name | = | OctetString4 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 |

Ce type a une longueur de quatre octets.

5.3.1.5.4 OctetString8

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 33 |
| 2 | Data type Name | = | OctetString8 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 8 |

Ce type octet string (chaîne d'octets) a une longueur de huit octets.

5.3.1.5.5 OctetString16

CLASS: Type de données

ATTRIBUTS:

- | | | | |
|-----|------------------------------|---|---------------|
| 1 | Data type Numeric Identifier | = | 34 |
| 2 | Data type Name | = | OctetString16 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 16 |

Ce type a une longueur de 16 octets.

5.3.1.6 Types Pointer (pointeurs)

5.3.1.6.1 Interface pointer (pointeur d'interface)

CLASS: Type de données

ATTRIBUTS:

- | | | |
|-----|------------------|-------------------|
| 2 | Data type Name = | Interface Pointer |
| 3 | Format | = FIXED LENGTH |
| 4.1 | Octet Length | = 4 |

Ce type de données définit un type de données de longueur fixe égale à 4 octets.

5.3.1.6.2 LPWSTR

CLASS: Type de données

ATTRIBUTS:

- | | | |
|---|------------------|----------------|
| 2 | Data type Name = | LPWSTR |
| 3 | Format | = FIXED LENGTH |

4.1 Octet Length = 4

Ce type de données définit une référence à un UnicodeString.

5.3.1.7 Types Time (temps)

5.3.1.7.1 BinaryTime0

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 40 |
| 2 | Data type Name | = | BinaryTime0 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 2 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de deux octets. L'unité de temps pour ce type est de 10 µs.

5.3.1.7.2 BinaryTime1

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 41 |
| 2 | Data type Name | = | BinaryTime1 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 2 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de deux octets. L'unité de temps pour ce type est de 100 µs.

5.3.1.7.3 BinaryTime2

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 42 |
| 2 | Data type Name | = | BinaryTime2 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 2 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de deux octets. L'unité de temps pour ce type est de 1 ms.

5.3.1.7.4 BinaryTime3

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 1 | Data type Numeric Identifier | = | 43 |
| 2 | Data type Name | = | BinaryTime3 |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 2 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de deux octets. L'unité de temps pour ce type est de 10 ms.

5.3.1.7.5 BinaryTime4

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	44
2	Data type Name	=	BinaryTime4
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de quatre octets. L'unité de temps pour ce type est de 10 µs.

5.3.1.7.6 BinaryTime5

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	45
2	Data type Name	=	BinaryTime5
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de quatre octets. L'unité de temps pour ce type est de 100 µs.

5.3.1.7.7 BinaryTime6

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	46
2	Data type Name	=	BinaryTime6
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de quatre octets. L'unité de temps pour ce type est de 1 ms.

5.3.1.7.8 BinaryTime7

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	47
2	Data type Name	=	BinaryTime7
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de six octets. L'unité de temps pour ce type est de 1 ms.

5.3.1.7.9 BinaryTime8

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	48
2	Data type Name	=	BinaryTime8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de six octets. L'unité de temps pour ce type est de 10 μ s.

5.3.1.7.10 BinaryTime9

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	49
2	Data type Name	=	BinaryTime9
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	6

Ce type est un nombre binaire. Le bit de poids fort de l'octet de poids fort est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type binary time (temps binaire) a une longueur de six octets. L'unité de temps pour ce type est de 100 μ s.

5.3.1.7.11 TIME

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	non utilisé
2	Data type Name	=	TIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type de la CEI 61131-3 est un nombre binaire en complément à deux avec une longueur égale à quatre octets. L'unité de temps pour ce type est de 1 ms.

5.3.1.7.12 ITIME

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	non utilisé
2	Data type Name	=	ITIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

Cette extension de type de la CEI 61131-3 est un nombre binaire en complément à deux avec une longueur égale à deux octets. L'unité de temps pour ce type est de 1 ms.

5.3.1.7.13 FTIME

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	non utilisé
2	Data type Name	=	FTIME
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Cette extension de type de la CEI 61131-3 est un nombre binaire en complément à deux avec une longueur égale à quatre octets. L'unité de temps pour ce type est de 1 μ s.

5.3.1.7.14 LTIME

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	non utilisé
2	Data type Name	=	LTIME
3	Format	=	FIXED LENGTH

4.1 Octet Length = 8

Cette extension de type de la CEI 61131-3 est un nombre binaire en complément à deux avec une longueur égale à huit octets. L'unité de temps pour ce type est de 1 μ s.

5.3.1.7.15 NetworkTime

CLASS: Type de données

ATTRIBUTES:

1 Data type Numeric Identifier = 58
2 Data type Name = NetworkTime
3 Format = FIXED LENGTH
4.1 Octet Length = 8

Ce type de données est constitué de deux valeurs non signées qui expriment le temps réseau.

Le premier élément est un type de données Unsigned32 qui donne le temps réseau en secondes à partir du 1900.01.01 (1^{er} janvier 1900) 00:00:00(TUC) pour le temps réseau supérieur ou égal à 1984.01.01 (1^{er} janvier 1984) 00:00:00 (TUC) et inférieur à 2036.07.02 (2 juillet 2036) 06:28:16(TUC), ou en secondes à partir du 2036.07.02 (2 juillet 2036) 06:28:16(TUC) pour le temps réseau supérieur ou égal à 2036.07.02 (2 juillet 2036) 06:28:16(TUC).

Le second élément est un type de données Unsigned32 qui donne la partie fractionnaire de secondes en $1/2^{32}$ s).

5.3.1.7.16 NetworkTimeDifference

CLASS: Type de données

ATTRIBUTES:

1 Data type Numeric Identifier = 59
2 Data type Name = NetworkTimeDifference
3 Format = FIXED LENGTH
4.1 Octet Length = 8

Ce type de données est constitué d'une valeur entière (Integer) et d'une valeur non signée (unsigned) qui expriment la différence de temps réseau. Le premier élément est un type de données Integer32 qui fournit la différence de temps réseau en secondes. Le second élément est un type de données Unsigned32 qui donne la partie fractionnaire de secondes en $1/2^{32}$ s).

5.3.1.7.17 Multiple time constant 2 octets (Multiple de constante de temps de 2 octets)

CLASS: Type de données

ATTRIBUTES:

1 Data type Numeric Identifier = 118
2 Data type Name = T2
3 Format = FIXED LENGTH
4.1 Octet Length = 2

Données de temps comme multiple du temps d'échantillonnage constant T_a .

Comme dans le cas de Unsigned16 avec une plage restreinte de valeurs $0 \leq x \leq 32\,767$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 0. L'attribut de type de donnée est montré dans le Tableau 16.

Tableau 16 – Attribut de multiple de constante de temps de 2 octets

Type de données	Plage de valeurs	Résolution	Longueur
T2	$0 \leq i \leq 32\,767 \times T_a$	T_a	2 Octets

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié T_a . Le temps d'échantillonnage associé est exigé afin d'interpréter la valeur interne.

5.3.1.7.18 Multiple time constant 4 octets (Multiple de constante de temps de 4 octets)

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 119
- 2 Data type Name = T4
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

Données de temps comme multiple du temps d'échantillonnage constant T_a .

Comme dans le cas de Unsigned32 avec une plage restreinte de valeurs $0 \leq x \leq 4\,294\,967\,295$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 0. L'attribut de type de donnée est montré dans le Tableau 17.

Tableau 17 – Attribut de multiple de constante de temps de 4 octets

Type de données	Plage de valeurs	Résolution	Longueur
T2	$0 \leq i \leq 4\,294\,967\,295 \times T_a$	T_a	4 Octets

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié T_a . Le temps d'échantillonnage associé est exigé afin d'interpréter la valeur interne.

5.3.1.7.19 Fraction time constant 2 octets (Fraction de constante de temps de 2 octets)

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 120
- 2 Data type Name = D2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Données de temps comme fraction du temps d'échantillonnage constant T_a .

Comme dans le cas de Unsigned16 avec une plage restreinte de valeurs $0 \leq x \leq 32\,767$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 0. Valeur interprétée = Valeur interne * $T_a/16\,384$. L'attribut de type de donnée est montré dans le Tableau 18.

Tableau 18 – Attribut de fraction de constante de temps de 2 octets

Type de données	Plage de valeurs	Résolution	Longueur
D2	$0 \leq i \leq (2 \cdot 2^{-14}) \times T_a$	$2^{-14} \times T_a$	2 Octets

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié T_a . Le temps d'échantillonnage associé est exigé afin d'interpréter la valeur interne.

5.3.1.7.20 Reciprocal time constant 2 octets (Inverse de constante de temps de 2 octets)

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 120
- 2 Data type Name = D2
- 3 Format = FIXED LENGTH

4.1 Octet Length = 2

Données de temps comme multiple inverse d'un temps d'échantillonnage constant T_a .

Comme dans le cas de Unsigned16 avec une plage limitée de valeurs $1\text{E} \times \text{E}16\ 384$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 16 384. Valeur interprétée = $16\ 384 \times T_a / \text{valeur interne}$. L'attribut de type de donnée est montré dans le Tableau 19.

Tableau 19 – Codage de l'inverse de constante de temps de 2 octets

Type de données	Plage de valeurs	Résolution	Longueur
R2	$1 \times T_a \leq i \leq 16\ 384 \times T_a$	T_a	2 Octets

NOTE La valeur de ce paramètre de temps renvoie au temps d'échantillonnage constant supplémentaire spécifié T_a . Le temps d'échantillonnage associé est exigé afin d'interpréter la valeur interne.

5.3.1.8 Types de caractères VisibleString

5.3.1.8.1 UNICODE char

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 36
- 2 Data type Name = UnicodeChar
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Ce type est défini comme un seul caractère dans le type chaîne UNICODE.

5.3.1.8.2 VisibleString1

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 25
- 2 Data type Name = VisibleString1
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 1

Ce type est défini comme un seul caractère dans le type VisibleString ISO.

5.3.1.8.3 VisibleString2

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 26
- 2 Data type Name = VisibleString2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 2

Ce type contient deux éléments du type VisibleString.

5.3.1.8.4 VisibleString4

CLASS: Type de données

ATTRIBUTES:

- 1 Data type Numeric Identifier = 27
- 2 Data type Name = VisibleString4
- 3 Format = FIXED LENGTH

4.1 Octet Length = 4

Ce type contient quatre éléments du type VisibleString.

5.3.1.8.5 VisibleString8

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	28
2	Data type Name	=	VisibleString8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

Ce type contient huit éléments du type VisibleString.

5.3.1.8.6 VisibleString16

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	29
2	Data type Name	=	VisibleString16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	16

Ce type contient 16 éléments du type VisibleString.

5.3.2 Types String (chaîne)

5.3.2.1 BitString

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	14
2	Data type Name	=	Bitstring
3	Format	=	STRING
5.1	Octet Length	=	1 à n

Ce type chaîne est défini comme une série d'éléments BitString8.

5.3.2.2 CompactBooleanArray

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	37
2	Data type Name	=	CompactBooleanArray
3	Format	=	STRING
6.1	Octet Length	=	1

Dans ce type, chaque valeur de bit égale à 0 (zéro) représente la valeur booléenne FALSE et chaque valeur de bit égale à 1 représente la valeur booléenne TRUE.

5.3.2.3 CompactBCDArray

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	38
2	Data type Name	=	CompactBCDArray
3	Format	=	STRING
4.1	Octet Length	=	1

Ce type sert à condenser une série ordonnée de valeurs BCD (c'est-à-dire DCB), deux par octet, en une série ordonnée d'octets. Le premier octet contient la valeur BCD de poids fort dans le quartet de poids fort. Si le nombre de valeurs BCD est impair, le quartet de poids faible de l'octet final est mis à la valeur réservée "1111".

5.3.2.4 OctetString

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|-------------|
| 1 | Data type Numeric Identifier | = | 10 |
| 2 | Data type Name | = | OctetString |
| 3 | Format | = | STRING |
| 4.1 | Octet Length | = | 1 à n |

Un OctetString est une séquence ordonnée d'octets, numérotés de 1 à n. Pour les besoins du débat, l'octet 1 de la séquence est appelé le premier octet. La CEI 61158-6 définit l'ordre d'émission.

5.3.2.5 UNICODEString

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|---------------|
| 1 | Data type Numeric Identifier | = | 39 |
| 2 | Data type Name | = | UnicodeString |
| 3 | Format | = | STRING |
| 4.1 | Octet Length | = | 2 |

Ce type est défini comme étant le type "string" (chaîne) UNICODE.

5.3.2.6 VisibleString

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|---------------|
| 1 | Data type Numeric Identifier | = | 9 |
| 2 | Data type Name | = | VisibleString |
| 3 | Format | = | STRING |
| 4.1 | Octet Length | = | 1 à n |

Ce type est défini comme étant le type "string" (chaîne) ISO/CEI 646.

5.4 Spécification de service de l'ASE de type de données

Il n'y a pas de services opérationnels définis pour l'objet de type.

6 Spécification de modèle de communication

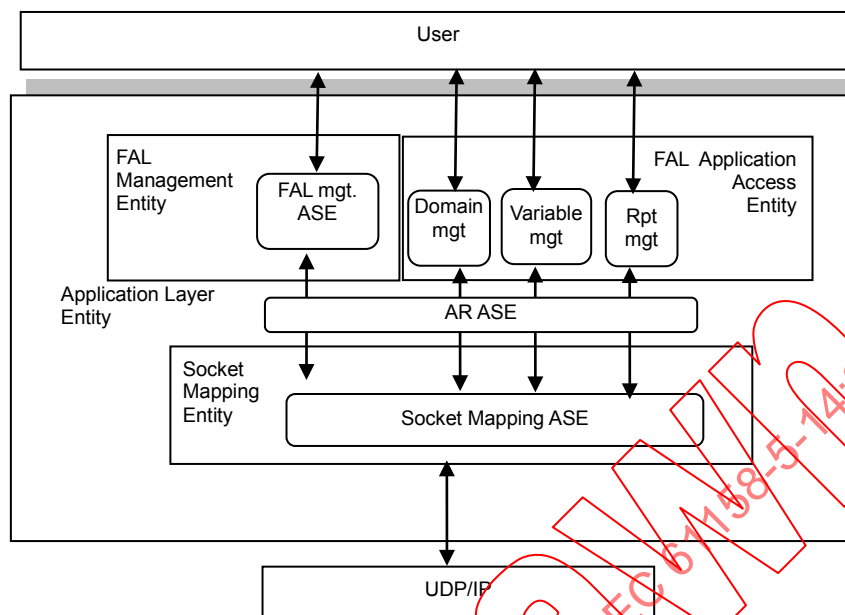
6.1 Généralités

La FAL AE (entité application de la couche application de bus de terrain) est constituée de la FAL Management Entity (FME), de l'Application Access Entity (AAE) et de la Socket Mapping Entity (SME), comme montré à la Figure 1.

La FAL Management Entity et les Services peuvent prendre en charge diverses opérations de gestion, y compris l'ASE FAL Management, l'ASE d'AR, etc.

L'AAE fournit une interface pour la communication de données entre un processus application utilisateur, qui se compose d'un ASE de domaine, un ASE de variable et un ASE d'événement.

La SME fournit une interface pour AAE, FME et UDP/IP, qui se compose de l'ASE Socket Mapping.



Légende

Anglais	Français
AR ASE	Elément de service application (ASE) de Relation d'applications (AR)
Rpt mgt	Gestion de rapports
Variable mgt	Gestion de variables
Domain mgt	Gestion de domaines
FAL mgt. ASE	ASE de gestion de couche application de bus de terrain (FAL)
Socket Mapping ASE	ASE mise en correspondance de ports
UDP/IP	Protocole datagramme utilisateur/Protocole internet
Socket Mapping Entity	Entité de mise en correspondance de ports
FAL Application Access Entity	Entité d'accès aux applications de la couche application de bus de terrain
FAL Management Entity	Entité de gestion de couche application de bus de terrain
Application Layer Entity	Entité de couche application
User	Utilisateur

Figure 1 – Entité de couche application

Pour les applications FRT, l'AAE et la FME peuvent aussi accéder directement à l'ECSME lorsque sont satisfaites les conditions suivantes :

- Les services d'AL appelés par l'AAE ou la FME doivent être des services non confirmés;
- la communication entre des points d'extrémité doit être restreinte dans un seul microsegment.

6.2 Les ASE

6.2.1 ASE "FAL Management" (gestion de FAL)

6.2.1.1 Vue d'ensemble

La gestion de FAL est utilisée pour intégrer plusieurs dispositifs du réseau en un système de communication harmonisé. La gestion de FAL prend en charge la fonction d'identification de dispositif, d'attribution d'adresse, de localisation d'objet, de synchronisation d'horloge et de gestion de liaison.

6.2.1.1.1 Identification de dispositif

Un dispositif peut être identifié par son étiquette de dispositif physique, son Device ID (c'est-à-dire: identificateur de dispositif), son numéro de redondance de dispositif (dispositifs mutuellement redondants).

Device ID

L'identificateur de dispositif identifie un dispositif unique. Un ID de dispositif unique est attribué par le fabricant à chaque dispositif individuel. Il est visible dans l'entité FAL Management, mais ne peut pas être modifié.

PD_Tag

Pour les besoins de la configuration d'un dispositif, chaque dispositif de terrain a une étiquette de dispositif physique (PD_Tag «physical device tag»). Dans un système de Type 14, PD_Tag est unique pour identifier un dispositif physique. Un dispositif mutuellement redondant a le même PD_Tag.

Numéro de redondance

Des dispositifs mutuellement redondants se distinguent les uns des autres par le numéro de redondance différent.

6.2.1.1.2 Attribution d'adresses

L'adresse IP de chaque dispositif peut être affectée de manière statique ou attribuée de manière dynamique par le biais du protocole de configuration de serveur dynamique (DHCP «Dynamic Host Configuration Protocol»). Une fois que l'adresse IP est attribuée, il convient de l'écrire dans la mémoire non volatile comme une information permanente jusqu'à ce que le dispositif soit configuré.

Lorsqu'un conflit de l'adresse IP avec un autre dispositif est détecté, il convient de changer l'état du dispositif local en *No Address* (aucune adresse), en attendant de réinitialiser son adresse IP.

6.2.1.1.3 Localisation d'objet

Un dispositif physique de Type 14 peut être localisé par l'adresse IP ou la PD_Tag. Une instance de FB dans un dispositif de Type 14 peut être localisée par la FB_Tag ou l'identificateur AppID de FB, qui sont uniques dans chaque dispositif, mais pas uniques dans l'ensemble du réseau. Chaque paramètre objet d'une instance de FB peut être localisé par le paramètre indice, ObjectID, qui est unique dans chaque instance de FB, mais pas unique dans l'ensemble du dispositif.

L'entité FAL Management fournit le service EM_DetectingDevice pour interroger l'adresse IP d'un dispositif en utilisant la PD_Tag. Elle fournit également le service EM_GetDeviceAttribute pour récupérer la PD_Tag ou autres informations pertinentes d'un dispositif physique.

La primitive "request" de service "EM_DetectingDevice" peut être VariableDistributed (distribuée par variable) dans un réseau avec un mode monodiffusion ou multidiffusion. Le dispositif physique qui reçoit cette primitive "request" vérifiera sa PD_Tag locale. Si la PD_Tag locale concorde avec celle contenue dans la primitive "request" de service "EM_DetectingDevice", il rapportera son adresse IP, son ID de dispositif en utilisant le service "EM_OnlineReply".

La primitive "request" de service "EM_GetDeviceAttribute" peut être délivrée à un réseau en utilisant le mode monodiffusion.

6.2.1.1.4 Ajout ou suppression d'un dispositif

Après la mise sous tension et l'initialisation, si un dispositif n'a pas d'adresse IP, il se met dans l'état *No Address* pour attendre l'attribution d'une adresse IP. Après attribution d'une adresse IP, le dispositif change d'état pour passer à *Unconfigured* (non configuré) et diffuse un message de dispositif en utilisant le service "EM_ActiveNotification".

Une fois que l'application d'utilisateur reçoit le message d'annonce de dispositif, elle peut configurer les attributs du dispositif en utilisant le service "EM_ConfiguringDevice". Après configuration, le dispositif peut fonctionner dans l'état *Configured* (configuré). Les informations de configuration sont écrites dans la mémoire non volatile comme informations permanentes jusqu'à ce que le dispositif soit reconfiguré.

Lorsqu'il est remis sous tension, le dispositif peut récupérer toutes les informations de configuration et fonctionner automatiquement dans l'état *Configured*.

6.2.1.1.5 Synchronisation de l'horloge

Il convient que l'heure de chaque dispositif soit synchronisée. Le protocole SNTP défini dans la RFC2030 ou celui défini dans la CEI 61588 peut être utilisé en fonction de l'application pratique.

À chaque synchronisation, le dispositif local envoie, à un intervalle de temps configuré, une primitive "request" de synchronisation d'horloge au serveur de temps et maintient la synchronisation avec l'heure système en fonction de la réponse donnée par le serveur de temps.

Les détails relatifs à la synchronisation de l'heure ne sont pas spécifiés dans la présente spécification (voir la RFC 2030 et la CEI 61588).

6.2.1.1.6 Gestion de liaison

Le chemin d'accès entre les FB est défini comme un objet liaison. L'objet liaison spécifie la relation de communication entre l'entrée/sortie des FB. Les objets liaisons sont configurés à l'aide d'un programme de configuration.

Par le biais d'un objet liaison configuré, un FB dans un dispositif peut déterminer où délivrer ses données de sortie ou bien où recevoir les données d'entrée de mise à jour.

6.2.1.1.7 Base d'objets de gestion

Tous les objets de gestion utilisés dans l'entité FAL Management sont organisés en une base d'objets de gestion (MOB «Management Object Base»). La MOB est un tableau bidimensionnel tel que montré dans le Tableau 20. Chaque objet a un identificateur unique ObjectID.

En tant que bloc fonctionnel spécial, la MOB a une valeur par défaut de son identificateur d'application AppID égale à 0 dans le but de lire/écrire les objets. Autrement dit,

l'identificateur d'application ApplD des autres instances de bloc fonctionnel a une valeur supérieure à 0.

Tableau 20 – Base d'objets de gestion

Objet	Object ID	Illustration
En tête de MIB	1	Objet en-tête de base d'objets de gestion d'un dispositif
Device Descriptor	2	Objet descripteur de dispositif
Time Synchronization	3	Objet synchronisation du temps
Max Response Time	4	Temps de réponse maximal d'un service confirmé
Communication Schedule Management	5	Objet gestion de programme de communication
Device Application information	6	Objet informations d'application de dispositif
FB Application information Header	7	En-tête d'informations d'application de bloc fonctionnel
Link Object Header	8	Objet en-tête de liaison
Domain Application Object Header	9	En-tête d'objet d'application de domaine
FRT Object Header	10	En-tête d'objet liaison temps réel rapide
.....		
FB Application information 1	2000	Informations d'application de bloc fonctionnel 1
FB Application information 2	2001	Informations d'application de bloc fonctionnel 2
.....	Nombre augmenté tour à tour	
Domain Application Object 1	4000	Objet d'application de domaine 1
Domain Application Object 2	4001	Objet d'application de domaine 2
.....	Nombre augmenté tour à tour	
Link Object 1	5000	Objet liaison 1
Link Object 2	5001	Objet liaison 2
.....	Nombre augmenté tour à tour	
FRT Link Object 1	7000	Objet liaison temps réel rapide 1
FRT Link Object 2	7001	Objet liaison temps réel rapide 2
.....	Nombre augmenté tour à tour	

6.2.1.2 Objets FAL Management

6.2.1.2.1 Classe MOB Header

La classe MOB Header (en-tête de base d'objets de gestion) représente le numéro de version de la MOB.

6.2.1.2.1.1 Modèle formel

La classe MOB Header est présentée comme suit:

ASE:	FAL MANAGEMENT ASE
CLASS:	MOB HEADER
CLASS ID:	Non utilisé

PARENT CLASS: TOP**ATTRIBUTES:**

1. (m) Attribut-clé: Object ID
2. (m) Attribut: MOB Revision Number

SERVICES:

1. (o) OpsService: Read

6.2.1.2.1.2 Attributs**Object ID**

Cet attribut identifie l'objet MOB Header dans la MOB. La valeur pour un objet MIB Header de dispositif est 1.

SMIB Revision Number

Cet attribut indique le numéro de révision de MOB qui est attribué par l'utilisateur.

6.2.1.2.1.3 Services**Read**

Le service permet à l'utilisateur de lire les attributs de la classe MOB Header.

6.2.1.2.2 Classe Device Descriptor

Cette classe spécifie les attributs de base d'un dispositif, à savoir, Device ID (identificateur de dispositif), Device Type (type de dispositif), Device Tag (étiquette de dispositif), IP address (adresse IP), etc. Par le biais des services de gestion de la FAL, le programme d'application utilisateur peut récupérer ou établir les attributs de base de ce dispositif.

6.2.1.2.2.1 Modèle formel**ASE:****FAL MANAGEMENT ASE****CLASS:****DEVICE DESCRIPTOR****CLASS ID:**

Non utilisé

PARENT CLASS:

TOP

ATTRIBUTES:

- | | | | |
|-----|-----|---------------|-----------------------------|
| 1. | (m) | Attribut-clé: | Object ID |
| 2. | (m) | Attribut: | Reserved |
| 3. | (m) | Attribut: | Application Type |
| 4. | (m) | Attribut-clé: | Device ID |
| 5. | (m) | Attribut: | PD_Tag |
| 6. | (m) | Attribut: | Active IP Address |
| 7. | (m) | Attribut: | Device Type |
| 8. | (m) | Attribut: | Status |
| 9. | (m) | Attribut: | Device Version |
| 10. | (m) | Attribut: | Annunciation Interval |
| 11. | (m) | Attribut: | Annunciation Version Number |
| 12. | (m) | Attribut: | Device Redundancy State |
| 13. | (m) | Attribut: | Device Redundancy Number |
| 14. | (m) | Attribut: | LAN Redundancy Port |
| 15. | (m) | Attribut: | Max Redundancy Number |
| 16. | (m) | Attribut: | Duplicate Tag Detected |

SERVICES:

- | | | | |
|----|-----|-------------|-----------------------|
| 1. | (o) | OpsService: | EM_DetectingDevice |
| 2. | (o) | OpsService: | EM_OnlineReply |
| 3. | (o) | OpsService: | EM_ConfiguringDevice |
| 4. | (o) | OpsService: | EM_GetDeviceAttribute |
| 5. | (o) | OpsService: | EM_SetDefaultValue |
| 6. | (o) | OpsService: | EM_ActiveNotification |

6.2.1.2.2.2 Attributs**Object ID**

Cet attribut identifie l'objet Device Descriptor dans la MOB. Sa valeur est 2 pour l'objet Device Descriptor.

Reserved

Ce champ est réservé.

Application type

Cet attribut indique le type d'application du dispositif local. Trois options sont proposées:

- 0: Applications régulières basées sur l'ISO/CEI 8802-3
- 1: Applications RT
- 2: Applications FRT

Device ID

Cet attribut spécifie le dispositif physique spécifique à un vendeur. Il est établi par le fabricant.

PD_Tag

Cet attribut spécifie le nom (étiquette) administré de site attribué au dispositif.

Active IP address

Cet attribut indique l'actuelle adresse IP opérationnelle.

Device type

Cet attribut spécifie le type de dispositif et sa capacité. Il est établi par le fabricant.

Status

Cet attribut indique le statut du dispositif local. Trois options sont proposées:

- 0: No address (Aucune adresse)
- 1: Unconfigured (non configuré)
- 2: Configured or operable (Configuré ou opérationnel)

Device version

Cet attribut est le numéro de version du dispositif. Il est attribué par le vendeur lorsqu'il est produit.

Annunciation interval

Cet attribut spécifie les millisecondes entre les messages d'annonce. Sa valeur par défaut est 15 000 (15 s).

Annunciation version number

Cet attribut est le numéro de version qui représente l'état composé des informations statiques dans les applications contenues dans le dispositif. Chaque fois qu'une modification est apportée aux données statiques de n'importe quelle application contenue dans le dispositif, ce numéro de version est incrémenté de 1. Sa valeur s'initialise à zéro si le dispositif est initialisé.

Device redundancy state

Cet attribut spécifie l'état réel de redondance du dispositif. 0 indique que le dispositif est actif alors que 1 indique que le dispositif est de secours. Si le dispositif ne participe pas à la redondance, sa valeur est 0xFF.

Device redundancy number

Cet attribut spécifie le numéro de redondance du dispositif. Une fois que le dispositif actif a une défaillance, le dispositif de secours de plus petit numéro devient en premier lieu un dispositif actif.

Max redundancy number

Cet attribut spécifie le numéro de redondance max.

LAN redundancy port

Cet attribut spécifie le port qui est utilisé pour échanger des données entre des dispositifs redondants.

Duplicate tag detected

L'attribut spécifie si, oui ou non, la PD_Tag du dispositif est en conflit avec une autre PD_Tag de dispositif dans le réseau. La valeur TRUE signale un conflit.

6.2.1.2.2.3 Services

EM_DetectingDevice

Ce service facultatif permet à l'utilisateur de trouver des informations de dispositif selon la PD_Tag de dispositif.

EM_OnlineReply

Ce service facultatif est utilisé pour répondre au dispositif qui envoie une demande de "EM_DetectingDevice".

EM_ConfiguringDevice

Cet service facultatif permet à l'utilisateur d'établir les attributs du dispositif.

EM_GetDeviceAttribute

Cet service facultatif permet à l'utilisateur de récupérer les attributs du dispositif.

EM_SetDefaultValue

Cet service facultatif permet à l'utilisateur d'effacer les attributs du dispositif.

EM_ActiveNotification

Ce service est utilisé pour un dispositif pour annoncer sa présence sur le réseau.

6.2.1.2.3 Classe Time Synchronization (synchronisation du temps)

Afin de mettre en œuvre la synchronisation d'horloge de dispositif dans le réseau, chaque dispositif maintient une Current Time (Heure courante). Par le biais de la synchronisation de l'heure, l'heure locale courante peut être synchronisée avec le serveur de temps avec la précision de synchronisation demandée.

6.2.1.2.3.1 Modèle formel

Cette classe est présentée comme suit:

ASE:	FAL MANAGEMENT ASE
CLASS:	TIME SYNCHRONIZATION
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut: Réserve

3.	(m)	Attribut:	Primary Time Server
4.	(m)	Attribut:	Secondary Time Server
5.	(m)	Attribut:	Time Request Timeout
6.	(m)	Attribut:	Time Request Interval
7.	(m)	Attribut:	Capable Time Sync Class
8.	(m)	Attribut:	Target Time Sync Class
9.	(m)	Attribut:	Current Time
10.	(m)	Attribut:	Standard Time Difference

SERVICES:

1.	(o)	OpsService:	Read
2.	(o)	OpsService:	Write
3.	(o)	OpsService:	Service Clock Synchronization (synchronisation d'horloge) défini dans la RFC 2030 ou la CEI 61588.

6.2.1.2.3.2 Attributs

Object ID

Cet attribut identifie l'objet Time Synchronization dans la MOB. Sa valeur est 3 pour l'objet Time Synchronization.

Reserved

Ce champ est réservé.

Primary time server

Cet attribut spécifie l'adresse IP du serveur de temps primaire (Primary Time Server).

Secondary time server

Cet attribut spécifie l'adresse IP du serveur de temps secondaire (Secondary Time Server).

Time request timeout

Cet attribut spécifie le temps (en microsecondes) du client de temps à attendre la demande provenant du serveur de temps

Time request interval

Cet attribut spécifie la durée (en secondes) du client de temps pour émettre la demande de synchronisation

Capable time sync class

Cet attribut spécifie la précision prise en charge par le client de temps pour la synchronisation du temps.

- 0: Aucune exigence de précision
- 1: Synchronisation de temps d'une précision <1 s
- 2: Synchronisation de temps d'une précision <100 ms
- 3: Synchronisation de temps d'une précision <10 ms
- 4: Synchronisation de temps d'une précision <1 ms
- 5: Synchronisation de temps d'une précision <100 µs
- 6: Synchronisation de temps d'une précision <10 µs
- 7: Synchronisation de temps d'une précision <1 µs
- 8: Synchronisation de temps d'une précision <100 ns
- 9: Synchronisation de temps d'une précision <10 ns

Target time sync class

Cet attribut indique la précision souhaitée prise en charge par le client de temps pour la synchronisation du temps. Il peut être établi par l'application utilisateur.

- 0: aucune exigence de précision
- 1: Synchronisation de temps d'une précision <1 s;
- 2: Synchronisation de temps d'une précision <100 ms;
- 3: Synchronisation de temps d'une précision <10 ms;
- 4: Synchronisation de temps d'une précision <1 ms;
- 5: Synchronisation de temps d'une précision <100 µs;
- 6: Synchronisation de temps d'une précision <10 µs;
- 7: Synchronisation de temps d'une précision <1 µs
- 8: Synchronisation de temps d'une précision <100 ns
- 9: Synchronisation de temps d'une précision <10 ns

Current time

Cet attribut indique l'heure d'horloge courante dans le dispositif local calculée à partir du 1^{er} janvier 1984. Son type de données est TimeOfDay.

Standard time difference

Cet attribut indique la différence temporelle entre l'heure système normalisée et l'heure courante. Son type de données est PrecisionTimeDifference sur 8 octets.

6.2.1.2.3.3 Services

Read

Le service permet à l'utilisateur de lire les attributs de la classe d'objets Clock Synchronization.

Write

Le service permet à l'utilisateur d'établir les attributs de la classe d'objets Clock Synchronization.

Service "Clock synchronization"

Les services nécessaires sont définis dans la RFC 2030 ou la CEI 61588.

6.2.1.2.4 Classe temps de réponse max de services confirmés

6.2.1.2.4.1 Modèle formel

ASE:		FAL MANAGEMENT ASE	
CLASS:		MAX RESPONSE TIME	
CLASS ID:		Non utilisé	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1.	(m)	Attribut-clé:	Object ID
2.	(m)	Attribut:	Réservé
3.	(m)	Attribut:	Max Response Time
SERVICES:			
1.	(o)	OpsService:	Read
2.	(o)	OpsService:	Write

6.2.1.2.4.2 Attributs

Object ID

Cet attribut identifie l'objet de classe Max Response Time de services confirmés dans la MOB. Sa valeur est 4.

Reserved

Ce champ est réservé.

Max response time

Cet attribut spécifie le temps de réponse maximal à attendre la réponse depuis que la primitive "request" a été envoyée pour un service confirmé. Son type de données est PrecisionTimeDifference sur 8 octets.

6.2.1.2.4.3 Services

Read

Le service permet à l'utilisateur de lire les attributs d'un objet temps de réponse max de service confirmé.

Write

Le service permet à l'utilisateur d'écrire les attributs d'un objet temps de réponse max de service confirmé.

6.2.1.2.5 Classe Communication scheduling management (gestion de programmation de communication)

Cette classe spécifie les informations utilisées dans une ECSME.

6.2.1.2.5.1 Modèle formel

ASE:	FAL MANAGEMENT ASE
CLASS:	COMMUNICATION SCHEDULE MANAGEMENT
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut: Communication Macrocycle
3. (m)	Attribut: NonPeriodic Data Transfer Offset
4. (m)	Attribut: Communication Macrocycle Version Number
SERVICES:	
1. (o)	OpsService: Read
2. (o)	OpsService: Write

6.2.1.2.5.2 Attributs

Object ID

Cet attribut identifie l'objet Communication Scheduling Management (gestion de programmation de communication) dans la MOB. Sa valeur est 5.

Communication macrocycle

Le type de données de "Communication Macrocycle" (macrocycle de communication) est PrecisionTimeDifference sur 8 octets ou PrecisionTimeDifference et peut être établi par l'utilisateur. La valeur par défaut de 0xFFFFFFFFFFFFFFFF indique que le "Communication Macrocycle" n'a pas été configuré.

Non-periodic data transfer offset

Cet attribut spécifie le décalage temporel de l'heure de début de NonPeriodicDataTransfer par rapport à l'heure de début d'un macrocycle de communication. Son type de données est PrecisionTimeDifference sur 8 octets. La valeur par défaut de 0xFFFFFFFFFFFFFFFF indique que cet attribut n'a pas été configuré.

Communication cycle version number

Cet attribut spécifie l'état de configuration d'un objet Communication Scheduling Management. Chaque fois que l'objet est modifié, la valeur de cet attribut est incrémentée de 1.

6.2.1.2.5.3 Services

Read

Le service permet à l'utilisateur de lire les attributs de la classe d'objets Communication Cycle Management.

Write

Le service permet à l'utilisateur d'établir les attributs de la classe d'objets Communication Cycle Management.

6.2.1.2.6 Classe Device application information

6.2.1.2.6.1 Modèle formel

ASE:		FAL MANAGEMENT ASE
CLASS:		DEVICE APPLICATION INFORMATION
CLASS ID:		Non utilisé
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Attribut-clé: Object ID
2.	(m)	Attribut: XDDL Version
SERVICES:		
1.	(o)	OpsService: Read

6.2.1.2.6.2 Attributs

Object ID

Cet attribut identifie l'objet Device Application Information dans la MOB. Sa valeur est 6.

XDDL Version

Cet attribut indique la version du document XDDL. Son type de données est Unsigned16, où l'octet de poids fort spécifie le numéro de version primaire alors que l'octet de poids faible indique le numéro de version secondaire.

6.2.1.2.6.3 Services

Read

Le service permet à l'utilisateur de lire les attributs de la classe d'objets Device Application Information.

6.2.1.2.7 Classe FB application information header

6.2.1.2.7.1 Modèle formel

ASE:		FAL MANAGEMENT ASE
CLASS:		FB APPLICATION INFORMATION HEADER
CLASS ID:		Non utilisé
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Attribut-clé: Object ID
2.	(m)	Attribut: Number of FB Application Information Object
3.	(m)	Attribut: First Number of FB Application Information Object
SERVICES:		
1.	(o)	OpsService: Read

6.2.1.2.7.2 Attributs

Object ID

Cet attribut identifie l'objet FB Application Information Object Header dans la MOB. Sa valeur est 7.

Number of FB application information object

Cet attribut indique le numéro de l'objet FB Application Information dans le dispositif local.

First Number of FB application information object

Cet attribut indique le premier numéro d'objet FB Application Information dans la MOB. Sa valeur est supérieure à l'ObjectID de l'en-tête d'objet d'application de domaine.

6.2.1.2.7.3 Services

Read

Le service facultatif permet à l'utilisateur de lire les attributs de la classe FB Application Information Object Header.

6.2.1.2.8 Classe Link object header

6.2.1.2.8.1 Modèle formel

ASE:		FAL MANAGEMENT ASE
CLASS:		LINK OBJECT HEADER
CLASS ID:		Non utilisé
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Attribut-clé: Object ID
2.	(m)	Attribut: Number of Link Object
3.	(m)	Attribut: First Number of Link Object
4.	(m)	Attribut: Number of Configured Link Object
5.	(m)	Attribut: Number of UnConfigured Link Object
SERVICES:		
1.	(o)	OpsService: Read

6.2.1.2.8.2 Attributs

Object ID

Cet attribut identifie l'objet Link Object Header dans la MOB. Sa valeur est 8.

Number of link object

Cet attribut indique le nombre d'objets liaisons dans la MOB.

First number of link object

Cet attribut indique le premier numéro d'objet liaison dans la MOB.

Number of configured link object

Cet attribut indique le nombre d'objets liaisons configurés par les utilisateurs.

Number of unconfigured link object

Cet attribut indique le nombre d'objets liaisons non configurés dans le dispositif local.

6.2.1.2.8.3 Service

Read

Le service facultatif permet à l'utilisateur de lire les attributs de la classe Link Object Header.

6.2.1.2.9 Classe Domain application information header

6.2.1.2.9.1 Modèle formel

ASE:	FAL MANAGEMENT ASE
CLASS:	DOMAIN APPLICATION INFORMATION HEADER
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut: Number of Domain Application Object
3. (m)	Attribut: First Number of Domain Application Object
4. (m)	Attribut: Number of Configured Domain Object
5. (m)	Attribut: Number of UnConfigured Domain Object
SERVICES:	
1. (o)	Ops Service: Read

6.2.1.2.9.2 Attributs

Object ID

Cet attribut identifie la classe Domain Application Information Object Header dans la MOB. Sa valeur est 9.

Number of domain application information object

Cet attribut indique le numéro de l'objet Domain Application Information dans le dispositif local.

First number of domain application object

Cet attribut indique le premier numéro d'objet Domain Application Information dans le dispositif local.

Number of configured domain object

Cet attribut indique le nombre d'objets Domain configurés dans le dispositif local.

Number of unconfigured domain object

Cet attribut indique le nombre d'objets Domain non configurés dans le dispositif local.

6.2.1.2.9.3 Services

Read

Ce service facultatif assure la lecture de l'attribut de la classe Domain Application Object Header.

6.2.1.2.10 Classe FRT Link object header

6.2.1.2.10.1 Modèle formel

ASE:	FAL MANAGEMENT ASE
CLASS:	FRT LINK OBJECT HEADER
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut: Number of FRT Link Object
3. (m)	Attribut: First Number of FRT Link Object
4. (m)	Attribut: Number of Configured FRT Link Object
5. (m)	Attribut: Number of UnConfigured FRT Link Object
SERVICES:	
1. (o)	OpsService: FRT Read

6.2.1.2.10.2 Attributs

Object ID

Cet attribut identifie l'objet Link Object Header dans la MOB. Sa valeur est 8.

Number of FRT link object

Cet attribut indique le nombre d'objets liaisons FRT (FRT Link Object) dans la MOB.

First number of FRT link object

Cet attribut indique le premier numéro d'objet liaison FRT dans la MOB.

Number of configured FRT link object

Cet attribut indique le nombre d'objets liaisons FRT (FRT link objects) configurés par les utilisateurs.

Number of unconfigured FRT link object

Cet attribut indique le nombre d'objets liaisons FRT (FRT Link Object) non configurés dans le dispositif local.

6.2.1.2.10.3 Service

FRTRead

Le service facultatif permet à l'utilisateur de lire les attributs de la classe FRT Link Object Header.

6.2.1.2.11 Classe FB application information

6.2.1.2.11.1 Modèle formel

ASE: FAL MANAGEMENT ASE

CLASS: FB APPLICATION INFORMATION

CLASS ID: Non utilisé

PARENT CLASS: TOP

ATTRIBUTES:

- | | | | |
|----|-----|---------------|-------------------------------|
| 1. | (m) | Attribut-clé: | Object ID |
| 2. | (m) | Attribut: | Réservé |
| 3. | (m) | Attribut: | FB Name |
| 4. | (m) | Attribut: | FB Type |
| 5. | (m) | Attribut: | Max Number of Instantiation |
| 6. | (m) | Attribut: | FB Execution Time |
| 7. | (m) | Attribut: | First Number of Instantiation |

SERVICES:

- | | | | |
|----|-----|--------------|------|
| 1. | (o) | Ops Service: | Read |
|----|-----|--------------|------|

6.2.1.2.11.2 Attributs

Object ID

Cet attribut identifie l'objet FB Application Information dans la MOB. Il convient que le numéro ObjectID des objets FB Application Information soit affecté en série.

Reserved

Cet attribut est un champ réservé.

FB name

Cet attribut sert à identifier un FB dans un dispositif local.

FB type

Cet attribut indique le type d'un FB.

Max number of instantiation

Cet attribut indique le nombre maximal d'instanciations d'un FB.

FB execution time

Cet attribut spécifie le temps d'exécution (en millisecondes pour les applications RT et en microsecondes pour les applications FRT) d'un FB.

First Number of Instantiation

Cet attribut définit le premier numéro qui peut être assigné pour l'instanciation de FB.

6.2.1.2.11.3 Services

Read

Ce service facultatif assure la lecture des attributs de l'objet FB Application Information.

6.2.1.2.12 Classe Link object

6.2.1.2.12.1 Modèle formel

ASE:		FAL MANAGEMENT ASE
CLASS:		LINK OBJECT
CLASS ID:		Non utilisé
PARENT CLASS:		TOP
ATTRIBUTES:		
1.	(m)	Attribut-clé: Object ID
2.	(m)	Attribut: LocalAppID
3.	(m)	Attribut: Local Object ID
4.	(m)	Attribut: RemoteAppID
5.	(m)	Attribut: RemoteObjectID
6.	(m)	Attribut: ServiceOperation
7.	(m)	Attribut: ServiceRole
8.	(m)	Attribut: RemoteIPAddress
9.	(m)	Attribut: SendTimeOffset
SERVICES:		
1.	(o)	Ops Service: Read
2.	(o)	Ops Service: Write

6.2.1.2.12.2 Attributs

Object ID

Cet attribut identifie le Link Object (objet liaison) dans la MOB. Il convient que le numéro ObjectID de Link Object soit affecté en série.

Local app ID

Cet attribut identifie l'instance FB locale.

Local object ID

Cet attribut identifie la variante locale de l'objet

Remote app ID

Cet attribut identifie l'instance FB distante.

RemoteObjectID

Cet attribut identifie la variante distante de l'objet

ServiceOperation

Cet attribut spécifie le service d'application devant être utilisé dans la relation de communication concernée:

- 0: liaison locale, aucun service d'application n'est utilisé.
- 1 à 17: le ServiceID des services d'application de Type 14 est utilisé.
- Autres: service non valide

ServiceRole

Cet attribut définit le rôle AREP du dispositif local dans le processus de communication.

- 0: SENDER, indiquant que le rôle AREP du dispositif local est CLIENT ou PUBLISHER.
- 1: RECEIVER, indiquant que le rôle AREP du dispositif local est SERVER ou SUBSCRIBER.
- Autres: L'objet liaison (Link Object) est non valide et 0xFF indique que l'objet liaison n'est pas configuré ou l'objet liaison a été supprimé.

RemoteIPAddress

Cet attribut identifie l'adresse IP du dispositif distant. Cet attribut peut être ignoré si l'objet local d'instanciation de FB et l'objet distant d'instanciation de FB se situent dans le même dispositif.

SendTimeOffset

Cet attribut définit le décalage temporel de l'instant où il convient d'envoyer le message pertinent par rapport à l'heure de début d'un macrocycle de communication. Cet attribut est valide lorsque Service ID est 0x0e (VARIABLEDISTRIBUTE) et ServiceRole est 0.

6.2.1.2.12.3 Service

Read

Ce service facultatif permet aux utilisateurs de lire les attributs du Link Object.

Write

Ce service facultatif permet aux utilisateurs de configurer les attributs du Link Object.

6.2.1.2.13 Classe Domain application information

6.2.1.2.13.1 Modèle formel

ASE:	FAL MANAGEMENT ASE
CLASS:	DOMAIN APPLICATION INFORMATION
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut: Domain Object ID
3. (m)	Attribut: ConfigurationStatus
4. (m)	Attribut: Reserved
5. (m)	Attribut: Domain Name
SERVICES:	
1. (o)	Ops Service: Read

6.2.1.2.13.2 Attributs

Object ID

Cet attribut identifie l'objet Domain Application Information dans la MOB. Il convient que le numéro ObjectID des objets Domain Application Information soit affecté en série.

Domain object ID

Cet attribut indique l'indice d'un Domain Object (objet domaine).

ConfigurationStatus

Cet attribut indique le statut de configuration d'un Domain Object. Il est TRUE si le Domain Object a été configuré.

Reserved

Cet attribut est un champ réservé.

Domain name

Cet attribut identifie un domaine local.

6.2.1.2.13.3 Services

Read

Ce service facultatif permet aux utilisateurs de lire les attributs de l'objet Domain Application Information.

6.2.1.2.14 Classe FRT Link object

6.2.1.2.14.1 Modèle formel

ASE:	FAL MANAGEMENT ASE
CLASS:	FRT LINK OBJECT
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut: Local Object ID
3. (m)	Attribut: RemoteObjectID
4. (m)	Attribut: ServiceOperation
5. (m)	Attribut: ServiceRole
6. (m)	Attribut: RemoteMACAddress
7. (m)	Attribut: SendTimeOffset
8. (m)	Attribut: ValidBitOffset
9. (m)	Attribut: ValidBitNumber
SERVICES:	
1. (o)	Ops Service: FRTRead
2. (o)	Ops Service: FRTWrite

6.2.1.2.14.2 Attributs

Object ID

Cet attribut identifie le FRT Link Object (objet liaison FRT) dans la MOB. Il convient que le numéro ObjectID de FRT Link Object soit affecté en série.

Local object ID

Cet attribut identifie la variante locale de l'objet

RemoteObjectID

Cet attribut identifie la variante distante de l'objet

ServiceOperation

Cet attribut spécifie le service d'application devant être utilisé dans la relation de communication concernée:

0: liaison locale, aucun service d'application n'est utilisé.

1 à 17: le ServiceID des services d'application de Type 14 est utilisé.

Autres: service non valide

ServiceRole

Cet attribut définit le rôle AREP du dispositif local dans le processus de communication.

0: SENDER, indiquant que le rôle AREP du dispositif local est CLIENT ou PUBLISHER.

1: RECEIVER, indiquant que le rôle AREP du dispositif local est SERVER ou SUBSCRIBER.

Autres: L'objet liaison (Link Object) est non valide et 0xFF indique que l'objet liaison n'est pas configuré ou l'objet liaison a été supprimé.

RemoteMACAddress

Cet attribut identifie l'adresse MAC du dispositif distant. Cet attribut peut être ignoré si l'objet local d'instanciation de FB et l'objet distant d'instanciation de FB se situent dans le même dispositif.

SendTimeOffset

Cet attribut définit le décalage temporel de l'instant où il convient d'envoyer le message pertinent par rapport à l'heure de début d'un macrocycle de communication. Cet attribut est valide lorsque Service ID est 0x12 (FRTVARIABLEDISTRIBUTE) et ServiceRole est 0.

ValidBitOffset

Cet attribut définit le décalage de bits de l'instant où il convient d'envoyer ou de recevoir le message pertinent par rapport à l'heure de début du champ DATA dans le service FRTVARIABLEDISTRIBUTE. Cet attribut est valide lorsque Service ID (identificateur de service) est 0x12 (FRTVARIABLEDISTRIBUTE).

ValidBitNumber

Cet attribut définit le nombre de bits de l'instant où il convient d'envoyer ou de recevoir le message pertinent par rapport à l'heure de début du champ DATA dans le service FRTVARIABLEDISTRIBUTE. Cet attribut est valide lorsque Service ID (identificateur de service) est 0x12 (FRTVARIABLEDISTRIBUTE).

6.2.1.2.14.3 Service

FRTRead

Ce service facultatif permet aux utilisateurs de lire les attributs du FRT Link Object (objet liaison FRT).

FRTWrite

Ce service facultatif permet aux utilisateurs de configurer les attributs du FRT Link Object.

6.2.2 ASE "application access entity" de la FAL

6.2.2.1 Attributs communs liés à l'accès

Password

Cet attribut spécifie le mot de passe pour le droit d'accès.

Access groups

Cet attribut permet de restreindre l'accès seulement aux clients qui appartiennent à l'un des groupes définis par l'attribut.

Cet attribut spécifie si l'objet appartient à un groupe d'utilisateurs (voir Tableau 21). Si l'un des bits est mis à 1, il indique un numéro de groupe auquel l'objet appartient.

Tableau 21 – Affectation de groupe d'accès

Bit	Signification
7	Groupe d'accès 1
6	Groupe d'accès 2
5	Groupe d'accès 3
4	Groupe d'accès 4
3	Groupe d'accès 5
2	Groupe d'accès 6
1	Groupe d'accès 7
0	Groupe d'accès 8

Access rights

Cet attribut définit le type d'accès défini pour le domaine (voir Tableau 22). Le droit d'accès correspondant est légitime pendant que le bit associé est mis.

Tableau 22 – Attribution de droits d'accès

Bit	Nom	Signification
7	R	Droit de lecture pour le mot de passe enregistré
6	W	Droit d'écriture pour le mot de passe enregistré
5	U	Droit d'utilisation pour le mot de passe enregistré
3	Rg	Droit de lecture pour les groupes d'accès
2	Wg	Droit d'écriture pour les groupes d'accès
1	Ug	Droit d'utilisation pour les groupes d'accès

6.2.2.2 Objet de domaine**6.2.2.2.1 Vue d'ensemble**

Un domaine représente une zone mémoire dont le contenu peut être des données ou un programme et la zone mémoire est représentée comme une séquence ordonnée d'octets. Un domaine peut être téléchargé vers l'amont ou vers l'aval, afin de maintenir l'intégrité, une seule opération de téléchargement amont ou aval à la fois étant autorisée pour un domaine.

L'ASE de domaine fournit les services suivants montrés dans le Tableau 23.

Tableau 23 – Services pour l'objet domaine

Nom de service	Service	Type	Objet
DomainDownload	Service de téléchargement aval pour le domaine	Confirmé	Objet Domaine
DomainUpload	Service de téléchargement amont pour le domaine	Confirmé	

6.2.2.2.2 Modèle formel

ASE:	FAL application ASE
CLASS:	DOMAIN
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTS:	
1. (m)	Attribut-clé: Object ID
2. (m)	Attribut-clé: Domain Name
3. (m)	Attribut: Max Octets

4.	(m)	Attribut:	Password
5.	(m)	Attribut:	Access Groups
6.	(m)	Attribut:	Access Rights
7.	(m)	Attribut:	Local Address
8.	(m)	Attribut:	Domain State
9.	(m)	Attribut:	Last State
10.	(m)	Attribut:	Used Application Counter

SERVICES:

1.	(o)	Ops Service:	DomainDownload
2.	(o)	Ops Service:	DomainUpload

6.2.2.2.3 Attributs

Object ID

Cet attribut est l'identificateur d'objet.

Domain name

Cet attribut spécifie le nom de l'objet domaine.

Max octets

Cet attribut spécifie la taille maximale du domaine en octets.

Password

Voir 6.2.2.1.

Access groups

Voir 6.2.2.1.

Access rights

Voir 6.2.2.1.

Local address

Cet attribut est l'adresse localement pertinente du domaine. La valeur de 0xFFFFFFFF indique qu'aucune adresse locale n'est disponible.

Domain state

Cet attribut spécifie les états suivants du domaine.

- 0: EXISTENT
- 1: DOWNLOADING
- 2: UPLOADING
- 3: READY
- 4: IN-USE

Last state

Cet attribut spécifie l'état du domaine avant le téléchargement vers l'amont ou vers l'aval.

Used application counter

Cet attribut spécifie le nombre de programmes utilisant ce domaine. Si la valeur de cet attribut est supérieure à 0, cela indique que le domaine est en cours d'utilisation. Le domaine ne peut pas être retéléchargé vers l'aval tant qu'il est en cours d'utilisation.

6.2.2.2.4 Services

DomainDownload

Ce service facultatif permet au client de télécharger vers l'aval un domaine.

DomainUpload

Ce service facultatif permet au client de télécharger vers l'amont un domaine.

6.2.2.3 Objet Report (rapport)

6.2.2.3.1 Vue d'ensemble

L'objet rapport est utilisé principalement pour que l'utilisateur envoie des messages importants d'un seul dispositif vers un certain nombre d'autres dispositifs (ou dans le mode diffusion, vers tous les autres dispositifs). C'est l'utilisateur qui définit les conditions qui déclenchent l'événement. Le processus application invoque le service EventReport pour notifier l'événement lorsque les conditions sont satisfaites. C'est également l'utilisateur qui confirme l'événement.

Par le biais du service Event Notification (notification d'événement), l'utilisateur émet l'Event Number (numéro d'événement) et l'Event Data (données d'événement). Par exemple: valeur de mesure, états et unité. La gestion de l'Event Number est parachevée par l'utilisateur. L'Event Notification peut représenter une alarme collective, dans laquelle les données peuvent contenir des informations indiquant quelle voie a déclenché l'alarme collective.

Le destinataire de l'Event Notification peut acquitter l'événement en utilisant le service AcknowledgeEventReport. La valeur de compteur est émise avec le service AcknowledgeEventReport. Le compteur sert à corréler l'EventReport et l'AcknowledgeEventReport.

Le destinataire de l'EventReport peut verrouiller ou déverrouiller l'EventReport en utilisant le service ReportConditionChanging.

L'ASE d'événement fournit trois services de gestion d'événements (voir Tableau 24).

Tableau 24 – Service pour objet rapport

Nom de service	Service	Confirmé/ Non confirmé	Objet
Event Notification	Notification d'événement	Non confirmé	ReportObject
AcknowledgeEventReport	Acquittement d'événement	Confirmé	
Alter Event Condition Monitor	Moniteur de changement de condition d'événement	Confirmé	

6.2.2.3.2 Modèle formel

ASE:	FAL Application ASE
CLASS:	EVENT
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTS:	
1. (m)	attribut-clé: Object ID
2. (m)	attribut: Longueur
3. (m)	attribut: Password
4. (m)	attribut: Access Groups
5. (m)	attribut: Access Rights
6. (m)	attribut: Local Address
7. (m)	attribut: Enabled