

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 © 2007 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 l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
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.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables 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 online 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 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC 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 l'IEC

La Commission Electrotechnique Internationale (IEC) 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 IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

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

La recherche avancée permet de trouver des publications IEC 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.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. 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 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 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

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

XC

ICS 25.040.40; 35.100.70

ISBN 978-2-8322-1489-3

**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 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.....	17
5.3 FAL defined data types	17
5.4 Data type ASE service specification.....	37
6 Communication model specification.....	38
6.1 General.....	38
6.2 ASEs.....	38
6.3 Application relationship.....	81
6.4 Summary of application layer services	84
Bibliography.....	86
Figure 1 – Application layer entity.....	38
Figure 2 – Received message processing procedure	60
Figure 3 – AR ASE conveys APDUs between AP	82
Table 1 – Management object base	40
Table 2 – Access group assignment.....	52
Table 3 – Access rights assignment.....	53
Table 4 – Services for domain object	53
Table 5 – Service for report object.....	55
Table 6 – FAL management entity services.....	62
Table 7 – EM_DetectingDevice service parameters	62
Table 8 – EM_OnlineReply service parameters.....	63
Table 9 – EM_GetDeviceAttribute service parameters	65
Table 10 – EM_ActiveNotification service parameters	67
Table 11 – EM_ConfiguringDevice service primitives	68

Table 12 – EM_SetDefaultValue service parameter	70
Table 13 – Parameters for domain download service	72
Table 14 – Parameters for domain upload service	74
Table 15 – EventReport service parameters	75
Table 16 – AcknowledgeEventReport service parameters	76
Table 17 – ReportConditionChanging service parameters	77
Table 18 – Read service parameters	78
Table 19 – Write service parameters	79
Table 20 – VariableDistribute service parameters	80
Table 21 – Summary of application layer services	84

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2007

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 provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 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 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 IEC 61784 series. 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 bilingual version (2014-06) corresponds to the English version, published in 2007-12.

This first edition and its companion parts of the IEC 61158-5 subseries cancel and replace IEC 61158-5:2003. This edition of this part constitutes a technical addition. This part and its Type 14 companion parts also cancel and replace IEC/PAS 62409, published in 2005.

This edition of IEC 61158-5 includes the following significant changes from the previous edition:

- a) deletion of the former Type 6 fieldbus for lack of market relevance;
- b) addition of new types of fieldbuses;
- c) partition of part 5 of the third edition into multiple parts numbered -5-2, -5-3, ...

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

FDIS	Report on voting
65C/475/FDIS	65C/486/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 of this standard has not been voted upon.

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

The committee has decided that the contents of this publication will remain unchanged until the maintenance result 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 The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

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

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 61158-5-14:2007

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

- 1) the FAL user at the boundary between the user and the application layer of the fieldbus reference model, and
- 2) 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) 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 standards 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 standard (including any amendments) applies.

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

IEC/TR 61158-1 (Ed.2.0), *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

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 – Part 1: The Basic Model*

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

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

ISO/IEC 8824, *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 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
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 |

- (1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.
- (2) 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.
- (3) 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.
- (4) 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.

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
 - a) 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.
 - b) 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.
 - c) 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).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
 - a) 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.
 - b) The label "OpsService" designates an operational service (1).
 - c) The label "MgtService" designates an management service (2).

- d) 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:

- 1) the parameter name;
- 2) the request primitive's input parameters;
- 3) the request primitive's output parameters;

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

- 4) the indication primitive's output parameters;
- 5) the response primitive's input parameters; and
- 6) 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

- 1) Parameter name,
- 2) request primitive,
- 3) indication primitive,
- 4) response primitive, and
- 5) 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) 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.
- b) 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 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 TimeDifference

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|----------------|
| 1 | Data type Numeric Identifier | = | 13 |
| 2 | Data type Name | = | TimeDifference |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 4 or 6 |

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The optional second element is an Unsigned16 data type that provides the difference in days.

5.3.1.3.11 TimeDifference with date indication

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|-------------------------------------|
| 1 | Data type Numeric Identifier | = | 53 |
| 2 | Data type Name | = | TimeDifference with date indication |
| 3 | Format | = | FIXED LENGTH |
| 4.1 | Octet Length | = | 6 |

This data type is composed of two elements of unsigned values that express the difference in time. The first element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds. The second element is an Unsigned16 data type that provides the difference in days.

5.3.1.3.12 TimeDifference without date indication

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	54
2	Data type Name	=	TimeDifference without date indication
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This data type is composed of one element of an unsigned value that express the difference in time. The element is an Unsigned32 data type that provides the fractional portion of one day in milliseconds.

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, 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 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.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Octet 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.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
				:				
				:				
Octet 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.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Octet 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.

SN = 0: positive numbers including zero

SN = 1: negative numbers.

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
				:				
				:				
Octet 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.

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

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

SN = 0: positive numbers including zero

SN = 1: negative numbers.

Data type	Range of values	Resolution	Length
E2	$-256 \cdot 2^{-7} \leq i \leq 256 \cdot 2^{-7}$	$2^{-7} = 0.0078125$	2 Octet

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Octet 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.

Data type	Range of values	Resolution	Length
C4	$-214748.3648 \leq i \leq 214748.3647$	$10^{-4} = 0.0001$	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.

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

Bit	8	7	6	5	4	3	2	1
Octet 1	Nibble 3				Nibble 2			
Octet 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 μ s.

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 μ s.

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 μ s.

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 µs.

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 µs.

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.

Data type	Range of values	Resolution	Length
T2	$0 \leq i \leq 32767 * T_a$	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.

Data type	Range of values	Resolution	Length
T2	$0 \leq i \leq 4294967295 * T_a$	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 * $T_a/16384$

Data type	Range of values	Resolution	Length
D2	$0 \leq i \leq (2 \cdot 2^{-14}) * T_a$	$2^{-14} * T_a$	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 values outside this range are set to 16384. Interpreted value = $16384 * T_a/\text{internal value}$.

☐ x ☐ 16384.

Data type	Range of values	Resolution	Length
R2	$1 * T_a \leq i \leq 16384 * T_a$	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.

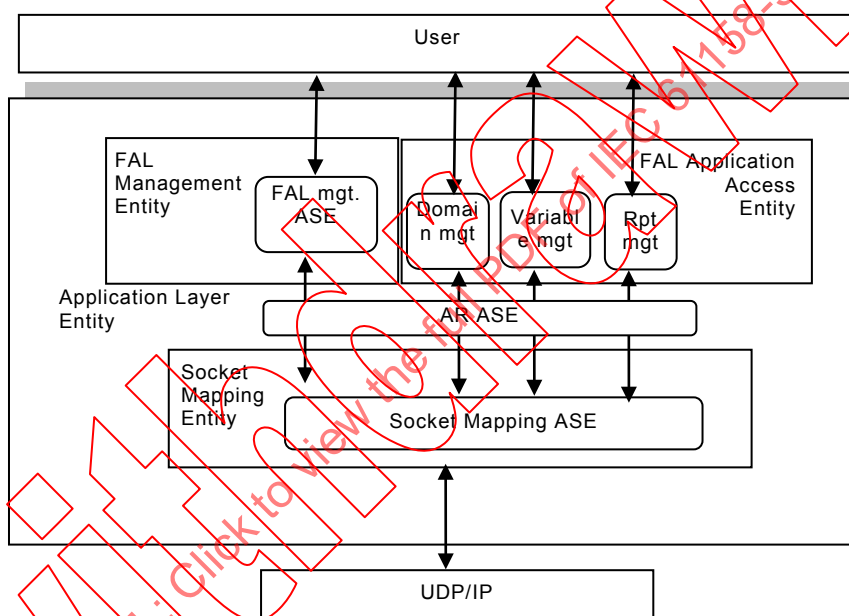


Figure 1 – Application layer entity

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 1. 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 1 – 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
.....		
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	
-------	--------------------------	--

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) Key Attribute: Device ID
4. (m) Attribute: PD_Tag
5. (m) Attribute: Active IP Address
6. (m) Attribute: Device Type
7. (m) Attribute: Status
8. (m) Attribute: Device Version
9. (m) Attribute: Annunciation Interval

10.	(m)	Attribute:	Annunciation Version Number
11.	(m)	Attribute:	Device Redundancy State
12.	(m)	Attribute:	Device Redundancy Number
13.	(m)	Attribute:	LAN Redundancy Port
14.	(m)	Attribute:	Max Redundancy Number
15.	(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.

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 µs
- 6: Time Synchronization precision <10 µs
- 7: Time Synchronization precision <1 µs

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 µs;
- 6: Time Synchronization precision <10 µs;
- 7: Time Synchronization precision <1 µs

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 6 octets TimeDifference.

6.2.1.2.3.3 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 4 octets of TimeDifference.

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 4 octets of TimeDifference and may be set by the user. The default value of 0xFFFFFFFF 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 4 octets of TimeDifference. The default value of 0xFFFFFFFF 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) OpService: 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 xonfigured 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 FB application information class**6.2.1.2.10.1 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

6.2.1.2.10.2 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) of an FB.

First Number of Instantiation

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

6.2.1.2.10.3 Services

Read

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

6.2.1.2.11 Link object class

6.2.1.2.11.1 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 |

6.2.1.2.11.2 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 0 (VARIABLEDISTRIBUTE) and ServiceRole is 0.

6.2.1.2.11.3 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.

6.2.1.2.12 Domain application information class**6.2.1.2.12.1 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

6.2.1.2.12.2 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.

6.2.1.2.12.3 Services

Read

This optional service permits users to read the attributes of Domain Application Information 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 2. If one of the bits is set to 1, it indicates a group number to which the object belongs.

Table 2 – 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 3. The corresponding access right is permitted while related bit has been set.

Table 3 – 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 4.

Table 4 – 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: FAL application ASE

CLASS: DOMAIN

CLASS ID: Not Used

PARENT CLASS: TOP

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 5.

Table 5 – 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: FAL Application ASE

CLASS: EVENT

CLASS ID: Not Used

PARENT CLASS: TOP

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.

- a) Mapping the application layer PDU to UDP protocol;
- b) Delivering the Application Layer PDUs into corresponding unsent buffer or queue according to the ServiceID identifier encapsulated in the PDUs.
- c) Providing overtime diagnosis and control for the confirmed services and returning positive or negative response to the corresponding entities.
- d) Providing priority management of application layer services.
- e) 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.

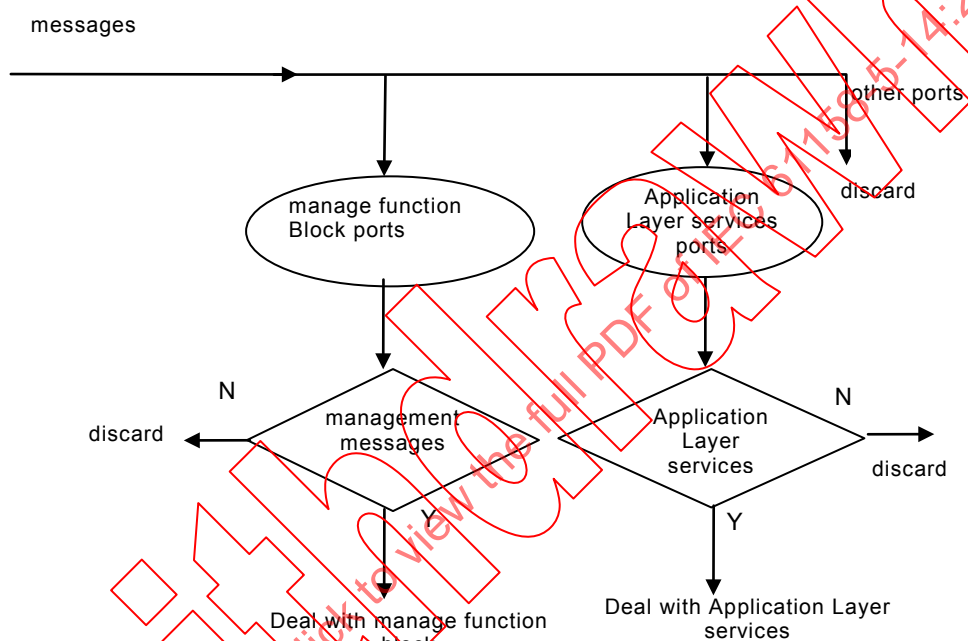


Figure 2 – Received message processing procedure

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 6.

Table 6 – 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 7.

Table 7 – 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 must 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 8.

Table 8 – 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 9.

Table 9 – 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 10.

Table 10 – 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 must be equal to the DeviceID of the device. When executing, if the device has already has a PD_Tag, it must 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 11.

Table 11 – 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 must 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 12.

Table 12 – 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 13.

Table 13 – 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 14.

Table 14 – 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 15.

Table 15 – 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 16.

Table 16 – 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 17.

Table 17 – 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 18.

Table 18 – 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 must 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 19.

Table 19 – 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 20.

Table 20 – 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.

6.3 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.

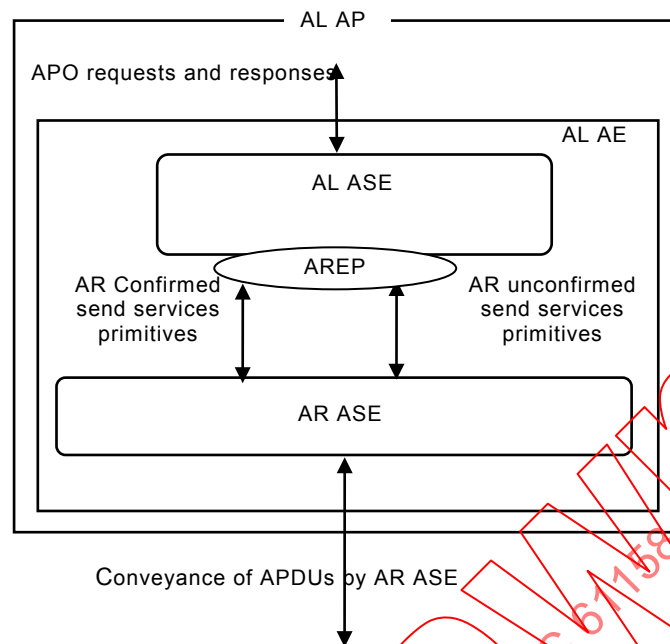


Figure 3 – AR ASE conveys APDUs between AP

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 21.

Table 21 – 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	SNTP Service		Unconfirmed	3	
16	ARP Service			1	
17	RARP, ICMP, IGMP, DHCP Service			4	
18	FTP, TFTP, HTTP Services			5	

IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2007

Withdrawn

Bibliography

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

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

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

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

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

IEC 61784-2, *Industrial communication networks – Profiles – Part 2: Additional fieldbus profiles for real-time networks based on ISO/IEC 8802-3*

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*

With IEC 61158-5-14:2007
IECNORM.COM : Click to view the full PDF

IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2007

SOMMAIRE

AVANT-PROPOS.....	90
INTRODUCTION.....	92
1 Domaine d'application	93
1.1 Vue d'ensemble.....	93
1.2 Spécifications.....	94
1.3 Conformité	94
2 Références normatives.....	94
3 Termes, définitions, symboles, abréviations et conventions	95
3.1 Termes de l'ISO/CEI 7498-1	95
3.2 Termes de l'ISO/CEI 8822	95
3.3 Termes de l'ISO/CEI 9545	95
3.4 Termes de l'ISO/CEI 8824	96
3.5 Définitions relatives à la couche Application de bus de terrain.....	96
3.6 Abréviations et symboles.....	99
3.7 Conventions	100
4 Concepts.....	103
5 Élément ASE de type de données.....	104
5.1 Vue d'ensemble.....	104
5.2 Définition formelle des objets de type de données	104
5.3 Types de données définis par la couche FAL	104
5.4 Spécification de service de l'élément ASE de type de données.....	126
6 Spécification de modèle de communication	126
6.1 Généralités.....	126
6.2 Éléments ASE.....	127
6.3 Relation d'applications.....	173
6.4 Résumé des services de couche Application	177
Bibliographie.....	178
Figure 1 – Entité de couche Application	127
Figure 2 – Procédure de traitement des messages reçus	151
Figure 3 – L'ASE d'AR achemine des APDU entre des AP	174
Tableau 1 – Base d'objets de gestion.....	130
Tableau 2 – Attribution de groupe d'accès	142
Tableau 3 – Attribution de droits d'accès	143
Tableau 4 – Services associés à l'objet de domaine	143
Tableau 5 – Service associé à l'objet de rapport	145
Tableau 6 – Services de l'entité FME	153
Tableau 7 – Paramètres du service EM_DetectingDevice	154
Tableau 8 – Paramètre du service EM_OnlineReply.....	155
Tableau 9 – Paramètres du service EM_GetDeviceAttribute	156
Tableau 10 – Paramètre du service EM_ActiveNotification.....	158
Tableau 11 – Primitives du service EM_ConfiguringDevice	160

Tableau 12 – Paramètres du service EM_SetDefaultValue.....	162
Tableau 13 – Paramètres du service Domain Download.....	163
Tableau 14 – Paramètres du service Domain Upload.....	165
Tableau 15 – Paramètres du service EventReport.....	167
Tableau 16 – Paramètres du service AcknowledgeEventReport	168
Tableau 17 – Paramètres du service ReportConditionChanging	169
Tableau 18 – Paramètres du service Read.....	170
Tableau 19 – Paramètres du service Write.....	171
Tableau 20 – Paramètres du service VariableDistribute	172
Tableau 21 – Résumé des services de couche Application	177

IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2007

COMMISSION ELECTROTECHNIQUE 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 Commission Electrotechnique Internationale (CEI) 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 les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de la CEI 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. Toutes divergences entre toutes Publications de la CEI et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) La CEI n'a prévu aucune procédure de marquage valant indication d'approbation et n'engage pas sa responsabilité pour les équipements déclarés conformes à une de ses Publications.
- 6) Il convient que tous les utilisateurs s'assurent qu'ils sont en possession de la dernière édition de la présente 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 la présente 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 Certains des types de protocole associés font l'objet d'une utilisation restreinte par les détenteurs de leurs droits de propriété intellectuelle. Dans tous les cas, l'engagement de renonciation partielle aux droits de propriété intellectuelle, pris par les détenteurs de ces droits, autorise l'utilisation d'un type de protocole de couche Liaison de données particulier avec des protocoles de couche physique et de couche Application dans les combinaisons de types explicitement spécifiées dans la série CEI 61784. L'utilisation des divers types de protocoles dans d'autres combinaisons peut nécessiter l'autorisation de leurs détenteurs de droits de propriété intellectuelle respectifs.

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 première édition et ses parties d'accompagnement de la sous-série CEI 61158-5 annulent et remplacent la CEI 61158-5:2003. La présente édition de cette partie constitue une addition technique. Cette partie et ses parties d'accompagnement pour le Type 14 annulent et remplacent également la CEI/PAS 62409, publiée en 2005.

Cette édition de la CEI 61158-5 inclut les modifications majeures suivantes par rapport à l'édition précédente:

- a) suppression du précédent bus de terrain de Type 6 en raison du manque d'adéquation au marché;
- b) ajout de nouveaux types de bus de terrain;
- c) division de cette partie en parties multiples numérotées -5-2,-5-3,...

La présente version bilingue (2014-06) correspond à la version anglaise monolingue publiée en 2007-12.

Le texte anglais de cette norme est issu des documents 65C/475/FDIS et 65C/486/RVD.

Le rapport de vote 65C/486/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.

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

Le comité a décidé que le contenu de la présente 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. A cette date, la publication sera:

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

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

La liste de toutes les parties de la série CEI 61158, publiées sous le titre général *Réseaux de communications industriels – Spécifications des bus de terrain*, est disponible sur le site web de la CEI.

INTRODUCTION

La présente partie de la CEI 61158 s'inscrit dans une série créée pour faciliter l'interconnexion des composants des systèmes d'automatisation. Elle renvoie à d'autres normes de l'ensemble défini par le modèle de référence de bus de terrain "à trois couches" décrit dans la CEI/TR 61158-1.

Le protocole d'application fournit le service d'application au moyen des services disponibles au niveau de la couche Liaison de données ou de la couche immédiatement inférieure. La présente norme définit les caractéristiques du service d'application que les applications de bus de terrain et/ou la gestion de système peuvent exploiter.

Dans l'ensemble de normes relatives aux bus de terrain, le terme "service" désigne une capacité abstraite fournie par une couche du modèle de référence de base de l'interconnexion des systèmes ouverts (*Open Systems Interconnection*, OSI) à la couche immédiatement supérieure. Ainsi, le service de couche Application défini dans la présente norme est un service architectural conceptuel, indépendant des divisions administratives et de mise en œuvre.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2007

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

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

1 Domaine d'application

1.1 Vue d'ensemble

La couche Application de bus de terrain (*Fieldbus Application Layer*, FAL) procure aux programmes de l'utilisateur un moyen d'accès à l'environnement de communication des bus de terrain. A cet égard, la FAL peut être considérée comme une "fenêtre entre programmes d'application correspondants".

La présente norme fournit des éléments communs pour les communications à temps critique ou non entre des programmes d'application dans un environnement et avec un matériel d'automation spécifiques aux bus de terrain de Type 14. Le terme "à temps critique" signale l'existence d'une fenêtre temporelle dans laquelle est exigée la réalisation d'une ou de plusieurs actions spécifiées, avec un niveau de certitude défini. La non-réalisation des actions spécifiées dans la fenêtre temporelle induit un risque de défaillance des applications qui demandent ces actions, avec les risques afférents pour l'équipement, les installations et éventuellement la vie humaine.

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

- a) de modèle abstrait visant à la définition des ressources d'application (objets) pouvant être manipulées par des utilisateurs utilisant un service FAL;
- b) d'événements et d'actions liées aux primitives du service;
- c) de paramètres associés à chaque événement et action de primitive, ainsi que de forme prise par ces paramètres, et
- d) d'interaction entre ces événements et ces actions, ainsi que de séquences valides desdits événements et actions.

La présente norme vise à définir les services mis en place pour:

- 1) l'utilisateur de FAL, à la frontière entre l'utilisateur et la couche Application du modèle de référence de bus de terrain; et
- 2) la Gestion des systèmes, à 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 de bus de terrain de Type 14, en conformité avec le modèle de référence de base de l'OSI (ISO/CEI 7498) et la structure de la couche Application de l'OSI (ISO/CEI 9545).

Les services et protocoles de couche FAL sont fournis par des entités d'application (*Application Entity*, AE) de couche FAL contenues dans les processus d'application. L'AE de la FAL se compose d'un jeu d'éléments de service d'application (*Application Service Element*, ASE) orientés objet et d'une entité de gestion de couche (*Layer Management Entity*, LME) qui gère l'AE. Les éléments ASE délivrent des services de communication agissant sur un ensemble de classes d'objets de processus d'application (*Application Process Object*, APO) associées. L'un des éléments ASE de couche FAL est un élément ASE de gestion qui fournit

un ensemble commun de services destinés à la gestion des instances des classes de couche FAL.

Quoique ces services spécifient, du point de vue des applications, les modalités d'émission et de remise des demandes et des réponses, ils ne comprennent pas de spécification du traitement que doivent en faire les applications demandeuse et répondeuse. En d'autres termes, les aspects comportementaux des applications ne sont pas définis; seule une définition des demandes et réponses que ces applications peuvent envoyer/recevoir est établie. Cela laisse une plus grande marge de manœuvre aux utilisateurs de la couche FAL dans la normalisation du comportement de ces objets. Outre ces services, la présente norme définit également certains services de soutien donnant accès à la couche FAL dans un but de commande de certains aspects de son fonctionnement.

1.2 Spécifications

La présente norme a pour principal objet de préciser les caractéristiques des services conceptuels de couche Application adaptés aux communications à temps critique; elle vise ainsi à compléter le modèle de référence de base OSI en guidant le développement de protocoles de couche Application destinés aux communications à temps critique.

Un objectif secondaire consiste à fournir des voies d'évolution à partir des protocoles de communication industriels antérieurs. 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 base pour des interfaces de programmation d'application formelles. Néanmoins, elle ne constitue pas une interface de programmation formelle, et toute interface de ce type devra traiter les questions de mise en œuvre non couvertes par cette spécification, y compris:

- a) les dimensions et l'ordre des octets de plusieurs paramètres de service multi-octet, et
- b) la corrélation des primitives associées (demande et confirmation, ou indication et réponse).

1.3 Conformité

La présente norme ne spécifie pas de mises en œuvre ni de produits particuliers, pas plus qu'elle ne limite les mises en œuvre des entités de couche Application dans les systèmes d'automation industriels.

Il n'existe pas de conformité de l'équipement à la présente norme de définition de service de couche Application. Au contraire, la conformité est obtenue par la 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 normes de référence suivantes 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).

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

CEI/TR 61158-1 (Ed.2.0), *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 1: Présentation et lignes directrices des séries CEI 61158 et CEI 61784*

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

ISO/IEC 646, *Information technology – ISO 7-bit coded character set for information interchange*(disponible en anglais seulement)

ISO/CEI 7498-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Partie 1: Le modèle de base*

ISO/CEI 7498-3, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Partie 3: Dénomination et adressage*

ISO/CEI 8822, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Définition du service de présentation*

ISO/CEI 8824, *Information Technology – Abstract Syntax notation One (ASN-1). Specification of basic notation* (disponible en anglais seulement)

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, 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 d'application
- d) élément de service d'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) application-association (association d'applications)
- b) application-context (contexte d'application)
- c) application-context-name (nom de contexte d'application)
- d) application-entity-invocation (invocation d'entité d'application)
- e) application-entity-type (type d'entité d'application)
- f) application-process-invocation (invocation de processus d'application)
- g) application-process-type (type de processus d'application)
- h) application-service-element (élément de service d'application)

i) application-control-service-element (élément de service de contrôle d'application)

3.4 Termes de l'ISO/CEI 8824

a) identificateur d'objet

b) type

3.5 Définitions relatives à la couche Application de bus de terrain

3.5.1

contrôle d'accès

contrôle des opérations de lecture et d'écriture portant sur un objet

3.5.2

chemin d'accès

association d'un nom symbolique à une variable dans le cadre d'une communication ouverte

3.5.3

macrocycle de communication

ensemble de cycles élémentaires nécessaires aux activités de communication configurées dans un macro-segment réseau

3.5.4

planification de la communication

algorithmes et comportement des opérations de transfert de données se déroulant de manière déterministe et reproductible

3.5.5

configuration (d'un système ou appareil)

étape de la conception d'un système consistant en la sélection des unités fonctionnelles, l'attribution de leurs emplacements et la définition de leurs interconnexions

3.5.6

cyclique

se dit d'un événement qui se reproduit de manière régulière et répétitive

3.5.7

instance du bloc de fonctions de destination

instance du bloc de fonctions qui reçoit les paramètres spécifiés

3.5.8

domaine

partie de la mémoire destinée au stockage de code ou de données

3.5.9

téléchargement de domaine

opération consistant à écrire des données dans un domaine

3.5.10

chargement de domaine

opération consistant à lire des données à partir d'un domaine

3.5.11

entité

élément particulier (personne, lieu, processus, objet, concept, association ou événement)

3.5.12**pont**

entité de relais DL qui effectue des synchronisations entre les liaisons (bus) et peut assurer des fonctions de stockage-transmission et d'acheminement sélectives dans le but de raccorder deux micro-segments réseau

3.5.13**identificateur**

mot de 16 bits associé à une variable système

3.5.14**index**

adresse d'un objet au sein d'un processus d'application

3.5.15**instance**

occurrence physique permettant d'identifier un objet parmi d'autres au sein d'une même classe d'objets

3.5.16**instanciation**

création d'une instance d'un type spécifié

3.5.17**informations de gestion**

informations mises à disposition sur le réseau à des fins de gestion du système de terrain

3.5.18**base d'informations de gestion**

liste organisée d'informations de gestion

3.5.19**mapping**

ensemble de valeurs ayant des correspondances définies avec les quantités ou les valeurs d'un autre ensemble

3.5.20**filtrage de messages**

décision concernant un message qui repose sur une règle spéciale

3.5.21**micro-segment**

partie d'un réseau sur laquelle est mise en œuvre une planification spéciale

3.5.22**position relative**

nombre d'octets enregistré à partir d'une position précise

3.5.23**phase**

partie écoulée d'un cycle, mesurée à partir d'un point d'origine fixe

3.5.24**interface de processus**

unité procédant à l'échange de données et au mapping d'informations entre un processus physique et une unité d'application

3.5.25

temps réel

capacité d'un système à fournir des résultats dans un délai limité

3.5.26

communication en temps réel

transfert de données en temps réel

3.5.27

Real-Time Ethernet (RTE)

réseau conforme à l'ISO/CEI 8802-3 qui inclut la communication en temps réel

NOTE 1 D'autres formes de communication peuvent être prises en charge, sous réserve que la communication en temps réel ne soit pas compromise.

NOTE 2 Cette définition s'applique, mais sans que cela soit limitatif, à l'ISO/CEI 8802-3. Elle peut s'appliquer à d'autres spécifications IEEE 802 (IEEE 802.11, par exemple).

3.5.28

planning

organisation temporelle de plusieurs opérations connexes

3.5.29

macrocycle de planification

délai de mise en application d'un planning particulier

3.5.30

instance du bloc de fonctions source

instance du bloc de fonctions qui envoie un paramètre spécifique

3.5.31

décalage temporel

laps de temps écoulé à partir d'une heure précise

3.6 Abréviations et symboles

AAE	Entité d'accès à l'application (<i>Application Access Entity</i>)
AE	Entité d'application (<i>Application Entity</i>)
AL	Couche Application (<i>Application Layer</i>)
ALME	Entité de gestion de couche Application (<i>Application Layer Management Entity</i>)
ALP	Protocole ALP (<i>Application Layer Protocol</i>)
APO	Objet d'application (<i>Application Object</i>)
AP	Processus d'application (<i>Application Process</i>)
APDU	Unité de données de protocole d'application (<i>Application Protocol Data Unit</i>)
API	Identificateur de processus d'application (<i>Application Process Identifier</i>)
AR	Relation d'applications (<i>Application Relationship</i>)
ARP	Protocole ARP (<i>Address Resolution Protocol</i>)
AREP	Point d'extrémité de relation d'applications (<i>Application Relationship End Point</i>)
ASE	Élément de service d'application (<i>Application Service Element</i>)
Cnf	Confirmation
CR	Relation de communication (<i>Communication Relationship</i>)
CREP	Point d'extrémité de relation de communication (<i>Communication Relationship End Point</i>)
CSMA/CD	Accès multiple par surveillance du signal et détection de collision (<i>Carrier Sense Multiple Access with Collision Detection</i>)
DD	Description d'appareil (<i>Device Description</i>)
DHCP	Protocole DHCP (<i>Dynamic Host Configuration Protocol</i>)
DL-	(comme préfixe) Liaison de données (<i>Data Link</i>)
DLCEP	Point d'extrémité de connexion de liaison de données (<i>Data Link Connection End Point</i>)
DLL	Couche Liaison de données (<i>Data Link Layer</i>)
DLE	Entité de liaison de données (<i>Data-Link Entity</i>)
DLM	Gestion de liaison de données (<i>Data Link Management</i>)
DLS	Service de liaison de données (<i>Data Link Service</i>)
DLSAP	Point d'accès au service de liaison de données (<i>Data Link Service Access Point</i>)
DLSDU	Unité de données de service de liaison de données (<i>DL-Service-Data Unit</i>)
ECSME	Entité de gestion de la planification des communications de Type 14 (<i>Type 14 Communication Scheduling Management Entity</i>)
Type 14	Système de transfert de fichiers reposant sur le protocole EPA (<i>Ethernet for Plant Automation</i>)
EM_	(comme préfixe) Gestion du Type 14 (<i>Ethernet Management_</i>)
ESME	Entité de mapping de ports de Type 14 (<i>Type 14 Socket Mapping Entity</i>)
FB	Bloc de fonctions (<i>Function Block</i>)
FBAP	Processus d'application du bloc de fonctions (<i>Function Block Application Process</i>)
Ind	Indication
IP	Protocole IP (<i>Internet Protocol</i>)
LLC	Commande de liaison logique (<i>Logical Link Control</i>)
LMP	Protocole LMP (<i>Link Management Protocol</i>)

MAC	Contrôle d'accès au support (<i>Medium Access Control</i>)
MAU	Unité de raccordement au support (<i>Medium Attachment Unit</i>)
MOB	Base d'objets de gestion (<i>Management Object Base</i>)
PAD	Bourrage (bits)
PDU	Unité de données de protocole (<i>Protocol Data Unit</i>)
P/S	Editeur/Abonné (<i>Publisher/Subscriber</i>)
Req	Demande (<i>Request</i>)
Rsp	Réponse (<i>Response</i>)
RTE	Real-Time Ethernet
RT-Ethernet	Real-Time Ethernet
SAP	Point d'accès au service (<i>Service Access Point</i>)
SDU	Unité de données de service (<i>Service Data Unit</i>)
SME	Entité de gestion de système (<i>System Management Entity</i>)
SNTP	Protocole SNTP (<i>Simple Network Time Protocol</i>)
TCP	Protocole TCP (<i>Transmission Control Protocol</i>)
UDP	Protocole UDP (<i>User Datagram Protocol</i>)

3.7 Conventions

3.7.1 Vue d'ensemble

La couche FAL se compose d'un ensemble d'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 des instances de la classe qui utilise les services d'ASE de gestion spécifiés à l'Article 5. La spécification de service définit les services fournis par l'ASE.

3.7.2 Conventions pour les définitions de classe

Les définitions de classe sont décrites à l'aide de modèles. Chaque modèle se compose d'une liste d'attributs associés à 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)	Key Attribute:	identificateur numérique
2	(o)	Key Attribute:	nom
3	(m)	Attribute:	nom d'attribut(valeurs)
4	(m)	Attribute:	nom d'attribut(valeurs)
4.1	(s)	Attribute:	nom d'attribut(valeurs)
4.2	(s)	Attribute:	nom d'attribut(valeurs)
4.3	(s)	Attribute:	nom d'attribut(valeurs)
5.	(c)	Constraint:	expression de la contrainte
5.1	(m)	Attribute:	nom d'attribut(valeurs)
5.2	(o)	Attribute:	nom d'attribut(valeurs)
6	(m)	Attribute:	nom d'attribut(valeurs)
6.1	(s)	Attribute:	nom d'attribut(valeurs)
6.2	(s)	Attribute:	nom d'attribut(valeurs)

SERVICES:

- | | | | |
|-----|-----|-------------|-----------------------------|
| 1 | (o) | OpsService: | nom du service |
| 2. | (c) | Constraint: | expression de la contrainte |
| 2.1 | (o) | OpsService: | nom du service |
| 3 | (m) | MgtService: | nom du service |

- (1) 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.
- (2) 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.
- (3) La rubrique "CLASS ID:" est un numéro qui identifie la classe spécifiée. Ce numéro est unique au sein de l'ASE FAL qui fournira les services pour cette classe. Lorsqu'il est qualifié par l'identité de son ASE FAL, 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) compris 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 compris entre 256 et 2048 sont alloués pour identifier les classes définies par l'utilisateur.
- (4) 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.

- (5) L'étiquette "ATTRIBUTES" indique que les entrées suivantes sont des attributs définis pour la classe.
- a) 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.
 - b) 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.
 - c) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point (period). 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).
- (6) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe.
- a) 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.
 - b) L'étiquette "OpsService" désigne un service opérationnel (1).
 - c) L'étiquette "MgtService" désigne un service de gestion (2).

- d) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par un point (period). 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 de service

3.7.3.1 Généralités

La présente norme emploie les conventions de description énoncées dans l'ISO/CEI 10731.

Le modèle de service, les primitives de service et les diagrammes séquentiels de temps utilisés sont des descriptions entièrement abstraites; ils ne représentent pas une spécification de 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 achèment 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 adaptés à une spécification de protocole, une spécification d'interface de programmation ou une 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 définis dans des tableaux jusqu'à la fin de la présente norme. Chaque tableau comporte jusqu'à six colonnes: une colonne pour le nom du paramètre de service et une colonne pour chacune des primitives et chacun des sens de transfert de paramètres utilisés par le service. Les six colonnes possibles sont:

- 1) le nom de paramètre;
- 2) les paramètres d'entrée de la primitive de demande;
- 3) les paramètres de sortie de la primitive de demande;

NOTE 2 Cette fonctionnalité est rarement utilisée. Sauf spécification contraire, les paramètres de la primitive de demande sont des paramètres d'entrée.

- 4) les paramètres de sortie de la primitive d'indication;
- 5) les paramètres d'entrée de la primitive de réponse; et
- 6) les paramètres de sortie de la primitive de confirmation.

NOTE 3 Les primitives de demande, d'indication, de réponse et de confirmation sont également connues comme primitives `requestor.submit` (demandeur.soumission), `acceptor.deliver` (accepteur.remise), `acceptor.submit` (accepteur.soumission) et `requestor.deliver` (demandeur.remise), respectivement (voir ISO/CEI 10731).

Les spécifications de service de 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 définis dans des tableaux. Chaque tableau comporte jusqu'à cinq colonnes:

- 1) le nom de paramètre;
- 2) la primitive de demande;
- 3) la primitive d'indication;
- 4) la primitive de réponse; et
- 5) la primitive de confirmation.

Un paramètre (ou un composant de celui-ci) est énuméré dans chaque ligne 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, en fonction de l'usage dynamique de l'utilisateur du service. Lorsque ce paramètre n'est pas fourni, une valeur par défaut est supposée;
- C le paramètre est conditionné à d'autres paramètres ou à l'environnement de l'utilisateur du service;
- (blanc/vide) le paramètre n'est jamais présent;
- S le paramètre est un élément sélectionné.

Certaines entrées sont par ailleurs qualifiées par des éléments entre parenthèses. Ces entrées peuvent être:

- a) une contrainte spécifique au paramètre:
 - "(=)" indique que le paramètre équivaut du point de vue de la sémantique au paramètre de la primitive de service située immédiatement à sa gauche dans le tableau.
- b) 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:

- d'interactions entre entités d'application par l'échange d'unités de données de protocole d'application de bus de terrain, et
- d'interactions entre un fournisseur de service 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 communication à contrainte temporelle au sein de la couche Application de bus de terrain.

NOTE La sous-série de normes CEI 61158-5 définit des 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. La mesure dans laquelle les procédures de service peuvent être mandatées dans les parties de la sous-série CEI 61158-5 est donc soumise à restrictions. Les aspects de protocole qui peuvent varier parmi différentes spécifications de protocole ou différentes mises en œuvre qui instantient 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 apparie des PDU de demande et de réponse peut être spécifié dans une norme de spécification de protocole de la CEI 61158-6, mais pas dans une norme de définition de services abstraits de la CEI 61158-5. De même, les méthodes de mise en œuvre locales par lesquelles un fournisseur de service ou un utilisateur de service apparie des primitives de demande et de confirmation ou des primitives d'indication et de réponse sont adaptées à une spécification de mise en œuvre ou une spécification d'interface de programmation, mais pas à une norme de service abstrait ou 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 conceptuel 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 la 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 la CEI/TR 61158-1, Article 9.

5 Éléments ASE de type de données

5.1 Vue d'ensemble

Une vue d'ensemble de l'ASE de type de données et des relations entre types de données est fournie dans la CEI/TR 61158-1, 10.1.

5.2 Définition formelle des objets de type de données

Le modèle utilisé pour décrire la classe de types de données du présent article est détaillé dans la CEI/TR 61158-1, 10.2. Il inclut la structure spécifique à l'élément ASE et la définition de ses attributs.

5.3 Types de données définis par la couche FAL

5.3.1 Types Fixed length (longueur fixe)

5.3.1.1 Types Boolean (booléens)

5.3.1.1.1 Boolean

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 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 un 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

ATTRIBUTES:

- | | | | |
|-----|----------------|---|--------------|
| 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 Integer16).

5.3.1.2 Types Bitstring (chaîne binaire)

5.3.1.2.1 BitString8

CLASS: Type de données

ATTRIBUTES:

- | | | | |
|-----|------------------------------|---|--------------|
| 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**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

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 Date**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 le plus 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 à 50 correspondent aux années 2000 à 2050; les valeurs 51 à 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus fort du nombre binaire; aucun bit de signe n'est inclus. Ce type Unsigned 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), et se terminant à 2151.06.06 (6 juin 2151), c'est-à-dire une plage totale de 65536 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 comprises entre le 1er 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 postérieurs au 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

ATTRIBUTES:

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 1er 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

ATTRIBUTES:

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

Ce type de données est constitué d'un seul élément de valeur unsigned (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 TimeDifference

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	13
2	Data type Name	=	TimeDifference
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4 ou 6

Ce type de données est constitué de deux éléments de valeurs unsigned (non signées) et exprime la différence de temps. Le premier élément est un type de données Unsigned32 qui fournit la partie fractionnaire d'un jour en millisecondes. Le deuxième élément facultatif est un type de données Unsigned16 qui donne la différence en jours.

5.3.1.3.11 TimeDifference with date indication (avec indication de date)

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	53
---	------------------------------	---	----

2	Data type Name	=	TimeDifference with date indication
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 la différence de temps. Le premier élément est un type de données Unsigned32 qui fournit la partie fractionnaire d'un jour en millisecondes. Le deuxième élément facultatif est un type de données Unsigned16 qui donne la différence en jours.

5.3.1.3.12 TimeDifference without date indication (sans indication de date)

CLASS: Type de données

ATTRIBUTES:

1	Data type Numeric Identifier	=	54
2	Data type Name	=	TimeDifference without date indication
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

Ce type de données est composé d'un élément d'une valeur unsigned (non signée) qui exprime la différence de temps. L'élément est un type de données Unsigned32 qui donne la partie fractionnaire d'un jour en millisecondes.

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 de temps sous la forme d'un nombre binaire en complément à deux d'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, 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 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

ATTRIBUTES:

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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de un octet. Dans ce type, les quatre bits de poids faible sont utilisés pour exprimer une valeur BCD 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

ATTRIBUTES:

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 de Float32 est celui défini par l'ANSI/IEEE 754 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

ATTRIBUTES:

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 de Float64 est celui défini par l'ANSI/IEEE 754 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

ATTRIBUTES:

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

Ce type entier est un nombre binaire en complément à deux d'une longueur de 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

ATTRIBUTES:

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

Ce type entier est un nombre binaire en complément à deux d'une longueur de 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

ATTRIBUTES:

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

Ce type entier est un nombre binaire en complément à deux d'une longueur de 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 est un nombre binaire en complément à deux d'une longueur de 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus fort du nombre binaire; aucun bit de signe n'est inclus. Ce type Unsigned 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus fort du nombre binaire; aucun bit de signe n'est inclus. Ce type Unsigned 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus fort du nombre binaire; aucun bit de signe n'est inclus. Ce type Unsigned 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).

En représentation en complément à deux, le bit de poids le plus fort (*Most Significant Bit*, MSB) est le bit qui suit le bit de signe (*Sign Bit*, SN) du premier octet.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs

Type de données	Plage de valeurs	Résolution	Longueur
N2	$-200 \% \leq i \leq (200 \cdot 2^{-14}) \%$	$2^{-14} = 0,0061 \%$	2 octets

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Octet 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

ATTRIBUTES:

- 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).

En représentation en complément à deux, le MSB est le bit qui suit le SN du premier octet.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
				:				
				:				
Octet 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

ATTRIBUTES:

- 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 pas automatiquement au bit 14 ou au bit 30, respectivement, mais est variable. Le bit de normalisation est codé en un paramètre complémentaire.

En représentation en complément à deux, le MSB est le bit qui suit le SN du premier octet.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Octet 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

ATTRIBUTES:

- 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 pas automatiquement au bit 14 ou au bit 30, respectivement, mais est variable. Le bit de normalisation est codé en un paramètre complémentaire.

En représentation en complément à deux, le MSB est le bit qui suit le SN du premier octet.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs

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

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
				:				
				:				
Octet 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

ATTRIBUTES:

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

Type de primitive avec des valeurs distinctes 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 en virgule fixe non signé".

NOTE 2 Ces types sont définis dans la CEI 61375.

Le nombre situé 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 transmise comme un entier non signé.

Type de données	Plage de valeurs	Résolution	Longueur
Unipolar2.16	$0 \% \leq i \leq (400 \cdot 2^{-14}) \%$	2^{-14}	2 octets

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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}
Partie entière		Partie fractionnaire													

5.3.1.4.4.17 Valeur en virgule fixe à 2 octets

CLASS: Type de données

ATTRIBUTES:

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

Valeur en virgule fixe linéaire avec sept binaires après la virgule décimale. 0 correspond à 0 (0x0), 128 correspond à 214 (0x4000).

En représentation en complément à deux, le MSB est le bit qui suit le SN du premier octet.

SN = 0: nombres positifs, y compris zéro

SN = 1: nombres négatifs

Type de données	Plage de valeurs	Résolution	Longueur
E2	$-256 \cdot 2^{-7} \leq i \leq 256 \cdot 2^{-7}$	$2^{-7} = 0,0078125$	2 octets

Bit	8	7	6	5	4	3	2	1
Octet 1	SN	2^7	2^6	2^5	2^4	2^3	2^2	2^1
Octet 2	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}

5.3.1.4.4.18 Valeur en virgule fixe à 4 octets

CLASS: Type de données

ATTRIBUTES:

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

Valeur en virgule fixe linéaire à quatre décimales. 0 correspond à 0 (0x0); 0,0001 correspond à 2° (0x0000 0001).

Comme dans le cas de Integer32, la pondération des bits a été réduite d'un facteur de 10000.

Type de données	Plage de valeurs	Résolution	Longueur
C4	$-214748,3648 \leq i \leq 214748,3647$	$10^{-4} = 0.0001$	4 Octet

5.3.1.4.4.19 Séquence de bits à 2 octets

CLASS: Type de données

ATTRIBUTES:

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

Séquence de bits permettant de contrôler et de représenter des fonctions d'application. 16 variables Boolean sont combinées en deux octets.

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

5.3.1.4.4.20 Quartet à 4 octets

CLASS: Type de données

ATTRIBUTES:

- 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. Quatre quartets sont représentés en deux octets.

La définition du quartet n'est pas spécifiée.

Bit	8	7	6	5	4	3	2	1
Octet 1	Quartet 3				Quartet 2			
Octet 2	Quartet 1				Quartet 0			

5.3.1.5 Types de caractère OctetString

5.3.1.5.1 OctetString1

CLASS: Type de données

ATTRIBUTES:

- 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 de un octet.

5.3.1.5.2 OctetString2

CLASS: Type de données

ATTRIBUTES:

- 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

ATTRIBUTES:

- 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

ATTRIBUTES:

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

ATTRIBUTES:

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 (pointeur)

5.3.1.6.1 Interface pointer (pointeur d'interface)

CLASS: Type de données

ATTRIBUTES:

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

ATTRIBUTES:

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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 μ s.

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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 μ s.

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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 µs.

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 le plus fort de l'octet de poids le plus fort est toujours utilisé comme bit de poids le plus 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 µs.

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 d'une longueur de 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 d'une longueur de 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 d'une longueur de quatre octets. L'unité de temps pour ce type est de 1 µs.

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 d'une longueur de huit octets. L'unité de temps pour ce type est de 1 µs.

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 unsigned (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 (1er janvier 1900) 00:00:00 (TUC) pour le temps réseau supérieur ou égal à 1984.01.01 (1er 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 des 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 integer (entière) et d'une valeur unsigned (non signée) 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 des secondes en $1/2^{32}$ s.

5.3.1.7.17 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, la plage de valeurs est réduite à $0 \leq x \leq 32\,767$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 0.

Type de données	Plage de valeurs	Résolution	Longueur
T2	$0 \leq i \leq 32\,767 * T_a$	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 requis pour l'interprétation de la valeur interne.

5.3.1.7.18 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, la plage de valeurs est réduite à $0 \leq x \leq 4294967295$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 0.

Type de données	Plage de valeurs	Résolution	Longueur
T2	$0 \leq i \leq 4294967295 * T_a$	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 requis pour l'interprétation de la valeur interne.

5.3.1.7.19 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, la plage de valeurs est réduite à $0 \leq x \leq 32\,767$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 0. Valeur interprétée = valeur interne * $T_a/16384$.

Type de données	Plage de valeurs	Résolution	Longueur
D2	$0 \leq i \leq (2 \cdot 2^{-14}) * T_a$	$2^{-14} * T_a$	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 requis pour l'interprétation de la valeur interne.

5.3.1.7.20 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, la plage de valeurs est réduite à $1 \leq x \leq 16384$. Lors de l'interprétation, les valeurs internes situées à l'extérieur de cette plage sont mises à 16384. Valeur interprétée = $16384 \times T_a / \text{valeur interne}$.

Type de données	Plage de valeurs	Résolution	Longueur
R2	$1 * T_a \leq i \leq 16384 * T_a$	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 requis pour l'interprétation de la valeur interne.

5.3.1.8 Types de caractère 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 UNICODE String.

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 ISO VisibleString.

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 String (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, deux par octet, en une série ordonnée d'octets. Le premier octet contient la valeur BCD de poids le plus fort dans le quartet de poids le plus fort. Si le nombre de valeurs BCD est impair, le quartet de poids le plus 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 de transmission.

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 UNICODE String.

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'élément 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

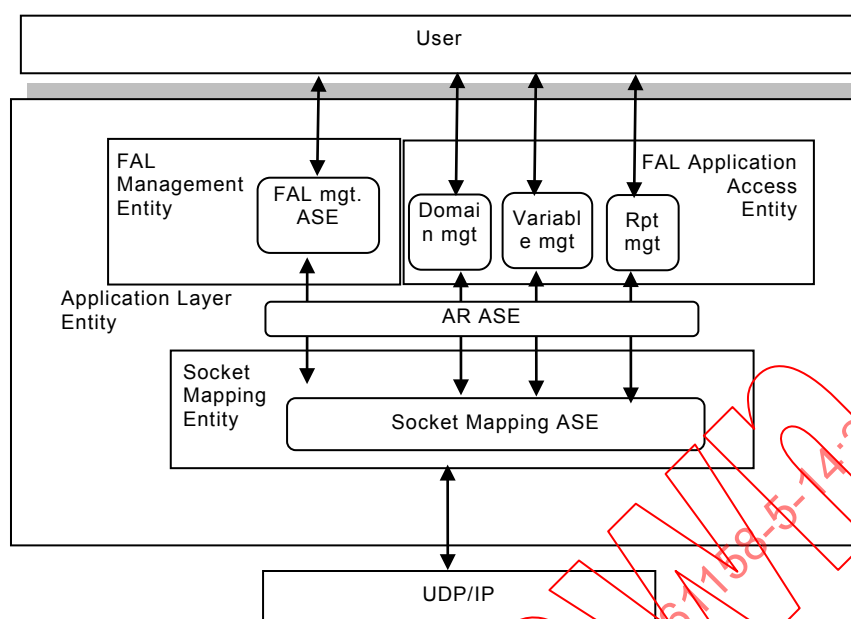
L'AE de la FAL est constituée de l'entité de gestion de FAL (*FAL Management Entity*, FME), de l'entité d'accès à l'application (*Application Access Entity*, AAE) et de l'entité de mapping de ports (*Socket Mapping Entity*, SME) (voir Figure).

La FME et les Services peuvent prendre en charge diverses opérations de gestion, y compris l'ASE de gestion de FAL, l'ASE d'AR, etc.

L'AAE fournit une interface pour la communication de données entre un processus d'application utilisateur, qui se compose d'un ASE de domaine, d'un ASE de variable et d'un ASE d'événement.

La SME fournit une interface pour AAE, FME et UDP/IP, qui se compose de l'ASE de mapping de ports.

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-14:2007



Légende

Anglais	Français
User	Utilisateur
FAL Management Entity	Entité FME
FAL mgt. ASE	ASE de gestion de couche FAL
Domain mgt	Gestion de domaines
Variable mgt	Gestion de variables
Rpt mgt	Gestion de rapports
FAL Application Access Entity	Entité AAE de couche FAL
Application Layer Entity	Entité ALE
AR ASE	ASE d'AR
Socket Mapping Entity	Entité SME
Socket Mapping ASE	ASE de mapping de ports

Figure 1 – Entité de couche Application

6.2 Eléments 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 appareils du réseau en un système de communication harmonisé. La gestion de FAL prend en charge la fonction d'identification d'appareil, d'attribution d'adresse, de localisation d'objet, de synchronisation d'horloge et de gestion de liaison.

6.2.1.1.1 Identification d'appareil

Un appareil peut être identifié par son étiquette d'appareil physique (PD_Tag), son identificateur d'appareil (Device ID) et son numéro de redondance d'appareil (appareils mutuellement redondants).

Device ID

L'identificateur d'appareil identifie un appareil unique. Un identificateur d'appareil unique est attribué par le fabricant à chaque appareil. Il est visible dans l'entité FME, mais ne peut pas être modifié.

PD_Tag

Pour les besoins de la configuration d'un appareil, chaque appareil de terrain porte une étiquette d'appareil physique (PD_Tag). Dans un système de Type 14, cette étiquette est unique et permet d'identifier un appareil physique. Un appareil mutuellement redondant possède la même étiquette d'appareil physique.

Numéro de redondance

Des appareils mutuellement redondants se distinguent les uns des autres par un numéro de redondance différent.

6.2.1.1.2 Attribution d'adresse

L'adresse IP de chaque appareil peut être affectée de manière statique ou attribuée de manière dynamique par le biais du protocole DHCP. 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 l'appareil soit reconfiguré.

Lorsqu'un conflit avec l'adresse IP d'un autre appareil est détecté, il convient de changer l'état de l'appareil local en *No Address* (aucune adresse), en attendant de réinitialiser son adresse IP.

6.2.1.1.3 Localisation d'objet

Un appareil physique de Type 14 peut être localisé par l'adresse IP ou la PD_Tag. Une instance du bloc de fonctions dans un appareil de Type 14 peut être localisée par la FB_Tag ou l'identificateur ApplD du bloc de fonctions, qui sont uniques dans chaque appareil, mais pas uniques dans l'ensemble du réseau. Chaque paramètre objet d'une instance du bloc de fonctions peut être localisé par le paramètre indice, ObjectID, qui est unique dans chaque instance du bloc de fonctions, mais pas unique dans l'ensemble de l'appareil.

L'entité FME fournit le service EM_DetectingDevice pour interroger l'adresse IP d'un appareil en utilisant la PD_Tag. Elle fournit également le service EM_GetDeviceAttribute pour récupérer la PD_Tag ou d'autres informations pertinentes concernant un appareil physique.

La primitive de demande de service EM_DetectingDevice peut être VariableDistributed (distribuée par variable) dans un réseau avec un mode monodiffusion ou multidiffusion. L'appareil physique qui reçoit cette primitive de demande vérifiera sa PD_Tag locale. Si la PD_Tag locale concorde avec celle contenue dans la primitive de demande de service EM_DetectingDevice, il rapportera son adresse IP et son Device ID en utilisant le service EM_OnlineReply.

La primitive de demande 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 appareil

Après la mise sous tension et l'initialisation, si un appareil n'a pas d'adresse IP, il passe à l'état *No Address* (aucune adresse) pour attendre l'attribution d'une adresse IP. Après attribution d'une adresse IP, l'appareil passe à l'état *Unconfigured* (non configuré) et diffuse un message d'annonce d'appareil en utilisant le service EM_ActiveNotification.

Une fois que l'application utilisateur reçoit le message d'annonce d'appareil, elle peut configurer les attributs de l'appareil en utilisant le service EM_ConfiguringDevice. Après configuration, l'appareil 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 l'appareil soit reconfiguré.

Lorsqu'il est remis sous tension, l'appareil peut récupérer toutes les informations de configuration et fonctionner automatiquement dans l'état *Configured* (configuré).

6.2.1.1.5 Synchronisation d'horloge

Il convient que l'heure de chaque appareil 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.

A chaque synchronisation, l'appareil local envoie, à un intervalle de temps configuré, une primitive de demande 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 RFC 2030 et CEI 61588).

6.2.1.1.6 Gestion de liaison

Le chemin d'accès entre les blocs de fonctions est défini comme un objet de liaison. L'objet de liaison spécifie la relation de communication entre l'entrée/sortie des blocs de fonctions. Les objets de liaison sont configurés à l'aide d'un programme de configuration.

Par le biais d'un objet de liaison configuré, un bloc de fonctions dans un appareil 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 FME sont organisés en une base d'objets de gestion (MOB). La MOB est un tableau bidimensionnel tel que montré dans le Tableau 1. Chaque objet a un identificateur unique ObjectID.

La MOB est bloc de fonctions spécial et, à ce titre, son identificateur d'application AppID possède une valeur par défaut égale à 0 dans le but de lire/écrire les objets. Autrement dit, l'identificateur d'application AppID des autres instances du bloc de fonctions a une valeur supérieure à 0.

Tableau 1 – Base d'objets de gestion

Objet	Object ID	Illustration
MIB Header	1	Objet d'en-tête de base d'objets de gestion d'un appareil
Device Descriptor	2	Objet de descripteur d'appareil
Time Synchronization	3	Objet de synchronisation du temps
Max Response Time	4	Temps de réponse maximal d'un service confirmé
Communication Schedule Management	5	Objet de gestion de planning de communication
Device Application information	6	Objet d'informations d'application d'appareil
FB Application information Header	7	En-tête d'informations d'application de bloc de fonctions
Link Object Header	8	En-tête d'objet de liaison
Domain Application Object Header	9	En-tête d'objet d'application de domaine
.....		
FB Application information 1	2000	Informations d'application de bloc de fonctions 1
FB Application information 2	2001	Informations d'application de bloc de fonctions 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 de liaison 1
Link Object 2	5001	Objet de liaison 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) Key Attribute: Object ID
2. (m) Attribute: 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 d'appareil 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

Ce 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 appareil, à savoir, Device ID (identificateur d'appareil), Device Type (type d'appareil), Device Tag (étiquette d'appareil), 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 cet appareil.

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) | Key Attribute: | Object ID |
| 2. | (m) | Attribute: | Reserved |
| 3. | (m) | Key Attribute: | Device ID |
| 4. | (m) | Attribute: | PD_Tag |
| 5. | (m) | Attribute: | Active IP Address |
| 6. | (m) | Attribute: | Device Type |
| 7. | (m) | Attribute: | Status |
| 8. | (m) | Attribute: | Device Version |
| 9. | (m) | Attribute: | Annunciation Interval |
| 10. | (m) | Attribute: | Annunciation Version Number |
| 11. | (m) | Attribute: | Device Redundancy State |
| 12. | (m) | Attribute: | Device Redundancy Number |
| 13. | (m) | Attribute: | LAN Redundancy Port |
| 14. | (m) | Attribute: | Max Redundancy Number |
| 15. | (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 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.

Device ID

Cet attribut spécifie l'appareil physique spécifique à un vendeur. Il est établi par le fabricant.

PD_Tag

Cet attribut spécifie le nom (étiquette) administré de site attribué à l'appareil.

Active IP address

Cet attribut indique l'adresse IP opérationnelle actuelle.

Device type

Cet attribut spécifie le type d'appareil et sa capacité. Il est établi par le fabricant.

Status

Cet attribut indique le statut de l'appareil 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 de l'appareil. 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 l'appareil. Chaque fois qu'une modification est apportée aux données statiques de n'importe quelle application contenue dans l'appareil, ce numéro de version est incrémenté de 1. Sa valeur s'initialise à zéro si l'appareil est initialisé.

Device redundancy state

Cet attribut spécifie l'état de redondance réel de l'appareil. 0 indique que l'appareil est l'appareil actif alors que 1 indique que l'appareil est un appareil de secours. Si l'appareil ne participe pas à la redondance, sa valeur est 0xFF.

Device redundancy number

Cet attribut spécifie le numéro de redondance de l'appareil. Lorsque l'appareil actif a une défaillance, l'appareil de secours de plus petit numéro devient l'appareil actif.

Max redundancy number

Cet attribut spécifie le numéro de redondance maximum.

LAN redundancy port

Cet attribut spécifie le port qui est utilisé pour échanger des données entre des appareils redondants.

Duplicate tag detected

Cet attribut spécifie si, oui ou non, la PD_Tag de l'appareil est en conflit avec celle d'un autre appareil du 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 sur un appareil en utilisant sa PD_Tag.

EM_OnlineReply

Ce service facultatif est utilisé pour répondre à l'appareil qui envoie une demande EM_DetectingDevice.

EM_ConfiguringDevice

Ce service facultatif permet à l'utilisateur d'établir les attributs de l'appareil.

EM_GetDeviceAttribute

Ce service facultatif permet à l'utilisateur de récupérer les attributs de l'appareil.

EM_SetDefaultValue

Ce service facultatif permet à l'utilisateur d'effacer les attributs de l'appareil.

EM_ActiveNotification

Ce service permet à un appareil d'annoncer sa présence sur le réseau.

6.2.1.2.3 Classe Time Synchronization

Afin de mettre en œuvre la synchronisation d'horloge d'appareil dans le réseau, chaque appareil maintient une Current Time (heure courante) locale. 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) 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: Service Clock Synchronization 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) d'attente de la demande provenant du serveur de temps.

Time request interval

Cet attribut spécifie le délai (en secondes, du client de temps) imparti pour transmettre 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 μ s
- 6: Synchronisation de temps d'une précision < 10 μ s
- 7: Synchronisation de temps d'une précision < 1 μ s

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 μ s
- 6: Synchronisation de temps d'une précision < 10 μ s
- 7: Synchronisation de temps d'une précision < 1 μ s

Current time

Cet attribut indique l'heure d'horloge courante dans l'appareil local calculée à partir du 1er janvier 1984. Son type de données est TimeOfDay.

Standard time difference

Cet attribut indique la différence de temps entre l'heure système normalisée et l'heure courante. Son type de données est TimeDifference sur 6 octets.

6.2.1.2.3.3 Services

Read

Ce service permet à l'utilisateur de lire les attributs de la classe d'objet Clock Synchronization.

Write

Ce service permet à l'utilisateur d'établir les attributs de la classe d'objets Clock Synchronization.

Clock synchronization service

Les services nécessaires sont définis dans la RFC 2030 ou la CEI 61588.

6.2.1.2.4 Classe Max Response Time de service confirmé

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) | 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 Attributs

Object ID

Cet attribut identifie l'objet de classe Max Response Time de service confirmé 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 de demande a été envoyée pour un service confirmé. Son type de données est TimeDifference sur 4 octets.

6.2.1.2.4.3 Services

Read

Ce service permet à l'utilisateur de lire les attributs d'un objet de temps de réponse maximal de service confirmé.

Write

Ce service permet à l'utilisateur d'écrire les attributs d'un objet de temps de réponse maximal de service confirmé.

6.2.1.2.5 Classe Communication Scheduling Management

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) 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 Attributs

Object ID

Cet attribut identifie l'objet Communication Scheduling Management dans la MOB. Sa valeur est 5.

Communication macrocycle

Le type de données de Communication Macrocycle est TimeDifference sur 4 octets et peut être établi par l'utilisateur. La valeur par défaut de 0xFFFFFFFF 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 TimeDifference sur 4 octets. La valeur par défaut de 0xFFFFFFFF 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

Ce service permet à l'utilisateur de lire les attributs de la classe d'objets Communication Cycle Management.

Write

Ce 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) Key Attribute: Object ID
2. (m) Attribute: 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 le plus fort spécifie le numéro de version primaire alors que l'octet de poids le plus faible indique le numéro de version secondaire.

6.2.1.2.6.3 Services**Read**

Ce 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) 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 Attributs**Object ID**

Cet attribut identifie l'objet FB Application Information 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 l'appareil 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**

Ce 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) 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 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 de liaison dans la MOB.

First number of link object

Cet attribut indique le premier numéro d'objet de liaison dans la MOB.

Number of xonfigured link object

Cet attribut indique le nombre d'objets de liaison configurés par les utilisateurs.

Number of unconfigured link object

Cet attribut indique le nombre d'objets de liaison non configurés dans l'appareil local.

6.2.1.2.8.3 Service

Read

Ce 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) 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 Attributs

Object ID

Cet attribut identifie la classe Domain Application Information Header dans la MOB. Sa valeur est 9.

Number of somain application informationobject

Cet attribut indique le numéro de l'objet Domain Application Information dans l'appareil local.

First number of domain application object

Cet attribut indique le premier numéro d'objet Domain Application Information dans l'appareil local.

Number of configured domain object

Cet attribut indique le nombre d'objets Domain configurés dans l'appareil local.

Number of unconfigured domain object

Cet attribut indique le nombre d'objets Domain non configurés dans l'appareil 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 FB Application Information

6.2.1.2.10.1 Modèle formel

ASE: FAL MANAGEMENT ASE

CLASS: FB APPLICATION INFORMATION

CLASS ID: Non utilisé

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

6.2.1.2.10.2 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 bloc de fonctions dans un appareil local.

FB type

Cet attribut indique le type d'un bloc de fonctions.

Max number of instantiation

Cet attribut indique le nombre maximal d'instanciations d'un bloc de fonctions.

FB execution time

Cet attribut spécifie le temps d'exécution (en millisecondes) d'un bloc de fonctions.

First Number of Instantiation

Cet attribut définit le premier numéro qui peut être assigné pour l'instanciation du bloc de fonctions.

6.2.1.2.10.3 Services

Read

Ce service facultatif assure la lecture des attributs de l'objet FB Application Information.

6.2.1.2.11 Classe Link Object

6.2.1.2.11.1 Modèle formel

ASE:	FAL MANAGEMENT ASE
CLASS:	LINK OBJECT
CLASS ID:	Non utilisé
PARENT CLASS:	TOP
ATTRIBUTES:	
1. (m) Key Attribute:	ObjectID
2. (m) Attribute:	LocalAppID
3. (m) Attribute:	Local ObjectID
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

6.2.1.2.11.2 Attributs

Object ID

Cet attribut identifie l'objet de liaison dans la MOB. Il convient que le numéro ObjectID de l'objet de liaison soit affecté en série.

Local app ID

Cet attribut identifie l'instance locale du bloc de fonctions.

Local object ID

Cet attribut identifie la variante locale de l'objet.

Remote app ID

Cet attribut identifie l'instance distante du bloc de fonctions.

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 de l'appareil local dans le processus de communication.

0: SENDER, indiquant que le rôle AREP de l'appareil local est CLIENT ou PUBLISHER.

1: RECEIVER, indiquant que le rôle AREP de l'appareil local est SERVER ou SUBSCRIBER.

Autres: L'objet de liaison est non valide et 0xFF indique que l'objet de liaison n'est pas configuré ou a été supprimé.

RemoteIPAddress

Cet attribut identifie l'adresse IP de l'appareil distant. Cet attribut peut être ignoré si l'objet local d'instanciation du bloc de fonctions et l'objet distant d'instanciation du bloc de fonctions se situent dans le même appareil.

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 0 (VARIABLEDISTRIBUTE) et ServiceRole est 0.

6.2.1.2.11.3 Service

Read

Ce service facultatif permet aux utilisateurs de lire les attributs de l'objet de liaison.

Write

Ce service facultatif permet aux utilisateurs d'écrire les attributs de l'objet de liaison.

6.2.1.2.12 Classe Domain Application Information

6.2.1.2.12.1 Modèle formel

ASE: FAL MANAGEMENT ASE
 CLASS: DOMAIN APPLICATION INFORMATION
 CLASS ID: Non utilisé
 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 |
|----|-----|--------------|------|

6.2.1.2.12.2 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 objet de domaine.

ConfigurationStatus

Cet attribut indique le statut de configuration d'un objet de domaine. Il est TRUE si le l'objet de domaine a été configuré.

Reserved

Cet attribut est un champ réservé.

Domain name

Cet attribut identifie un domaine local.

6.2.1.2.12.3 Services

Read

Ce service facultatif permet aux utilisateurs de lire les attributs de l'objet Domain Application Information.

6.2.2 ASE FAL Application Access Entity

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 2). Si l'un des bits est mis à 1, il indique un numéro de groupe auquel l'objet appartient.

Tableau 2 – Attribution 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 3). Le droit d'accès correspondant est légitime pendant que le bit associé est mis.

Tableau 3 – 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 Domain**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 chargé ou téléchargé; afin de maintenir l'intégrité, une seule opération de téléchargement ou de chargement à la fois est autorisée pour un domaine.

L'ASE de domaine fournit les services présentés dans le Tableau 4.

Tableau 4 – Services associés à l'objet de domaine

Nom de service	Service	Type	Objet
DomainDownload	Service de téléchargement pour le domaine	Confirmé	Objet Domain
DomainUpload	Service de chargement pour le domaine	Confirmé	

6.2.2.2.2 Modèle formel

ASE: FAL application ASE

CLASS: DOMAIN

CLASS ID: Non utilisé

PARENT CLASS: TOP

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 Attributs**Object ID**

Cet attribut est l'identificateur d'objet.

Domain name

Cet attribut spécifie le nom de l'objet de domaine.