

# 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 © 2014 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](mailto:info@iec.ch)  
[www.iec.ch](http://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](http://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](http://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](http://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](http://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](http://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](http://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](mailto: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](http://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](http://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](http://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](http://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](http://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](http://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](mailto:csc@iec.ch).



IEC 61158-5-14

Edition 3.0 2014-08

# 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 – Eléments  
de type 14**

INTERNATIONAL  
ELECTROTECHNICAL  
COMMISSION

COMMISSION  
ELECTROTECHNIQUE  
INTERNATIONALE

PRICE CODE  
CODE PRIX

**XD**

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-1739-9

**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 General.....                                                   | 7  |
| 1.2 Specifications.....                                            | 8  |
| 1.3 Conformance.....                                               | 8  |
| 2 Normative references .....                                       | 8  |
| 3 Terms, definitions, symbols, abbreviations and conventions ..... | 9  |
| 3.1 ISO/IEC 7498-1 terms .....                                     | 9  |
| 3.2 ISO/IEC 8822 terms .....                                       | 9  |
| 3.3 ISO/IEC 9545 terms .....                                       | 9  |
| 3.4 ISO/IEC 8824-1 terms .....                                     | 10 |
| 3.5 Fieldbus application-layer specific definitions .....          | 10 |
| 3.6 Abbreviations and symbols.....                                 | 12 |
| 3.7 Conventions .....                                              | 13 |
| 4 Concepts.....                                                    | 17 |
| 5 Data type ASE.....                                               | 17 |
| 5.1 Overview .....                                                 | 17 |
| 5.2 Formal definition of data type objects .....                   | 17 |
| 5.3 FAL defined data types.....                                    | 17 |
| 5.4 Data type ASE service specification .....                      | 39 |
| 6 Communication model specification.....                           | 39 |
| 6.1 General.....                                                   | 39 |
| 6.2 ASEs.....                                                      | 40 |
| 6.3 Application relationship .....                                 | 93 |
| 6.4 Summary of application layer services.....                     | 96 |
| Bibliography.....                                                  | 98 |
| Figure 1 – Application layer entity .....                          | 39 |
| Figure 2 – Received message processing procedure .....             | 64 |
| Figure 3 – AR ASE conveys APDUs between AP .....                   | 94 |
| Table 1 – Attribute of variable normalised 2 octet .....           | 26 |
| Table 2 – Encoding of variable normalised 2 octet .....            | 26 |
| Table 3 – Attribute of normalised 4 Octet.....                     | 26 |
| Table 4 – Encoding of normalised 4 Octet.....                      | 26 |
| Table 5 – Attribute of variable normalised 2 octet .....           | 27 |
| Table 6 – Encoding of variable normalised 2 octet .....            | 27 |
| Table 7 – Attribute of variable normalised 4 Octet.....            | 27 |
| Table 8 – Encoding of variable normalised 4 Octet .....            | 28 |
| Table 9 – Attribute of unipolar 2 octet .....                      | 28 |
| Table 10 – Encoding of unipolar 2 octet.....                       | 28 |
| Table 11 – Attribute of Fixed point value 2 Octet.....             | 29 |
| Table 12 – Encoding of Fixed point value 2 Octet .....             |    |

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Table 13 – Attribute of Fixed point value 4 Octet.....               | 29 |
| Table 14 – Encoding of Bit sequence 2 Octet.....                     | 30 |
| Table 15 – Encoding of Nibble 4 Octet.....                           | 30 |
| Table 16 – Attribute of multiple time constant 2 octets .....        | 35 |
| Table 17 – Attribute of multiple time constant 4 octets .....        | 36 |
| Table 18 – Attribute of fraction time constant 2 octets.....         | 36 |
| Table 19 – Encoding of reciprocal time constant 2 octets .....       | 36 |
| Table 20 – Management object base.....                               | 42 |
| Table 21 – Access group assignment.....                              | 56 |
| Table 22 – Access rights assignment .....                            | 57 |
| Table 23 – Services for domain object .....                          | 57 |
| Table 24 – Service for report object .....                           | 59 |
| Table 25 – FAL management entity services.....                       | 66 |
| Table 26 – EM_DetectingDevice service parameters.....                | 67 |
| Table 27 – EM_OnlineReply service parameters .....                   | 68 |
| Table 28 – EM_GetDeviceAttribute service parameters.....             | 69 |
| Table 29 – EM_ActiveNotification service parameters .....            | 71 |
| Table 30 – EM_ConfiguringDevice service primitives .....             | 72 |
| Table 31 – EM_SetDefaultValue service parameter .....                | 74 |
| Table 32 – Parameters for domain download service .....              | 76 |
| Table 33 – Parameters for domain upload service.....                 | 78 |
| Table 34 – EventReport service parameters.....                       | 79 |
| Table 35 – AcknowledgeEventReport service parameters .....           | 80 |
| Table 36 – ReportConditionChanging service parameters .....          | 81 |
| Table 37 – Read service parameters.....                              | 83 |
| Table 38 – Write service parameters.....                             | 84 |
| Table 39 – VariableDistribute service parameters .....               | 85 |
| Table 40 – FRTVariableDistribute service parameters.....             | 86 |
| Table 41 – FRTRead service parameters .....                          | 86 |
| Table 42 – FRTWrite service parameters .....                         | 87 |
| Table 43 – Parameters for Block Transmission Open service .....      | 89 |
| Table 44 – Parameters for Block Transmission Close service .....     | 90 |
| Table 45 – Parameters for BlockTransmit service .....                | 91 |
| Table 46 – Parameters for Block Transmission Heartbeat service ..... | 93 |
| Table 47 – Summary of application layer services.....                | 97 |

## INTERNATIONAL ELECTROTECHNICAL COMMISSION

### INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

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

#### FOREWORD

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

Attention is drawn to the fact that the 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 layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by their respective intellectual property right holders.

NOTE Combinations of protocol Types are specified in IEC 61784-1 and IEC 61784-2.

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 third edition cancels and replaces the second edition published in 2010. This edition constitutes a technical revision. The main changes with respect to the previous edition are listed below:

- corrections of the edit error;
- specification changes for CPF4;
- update of the requirements for all conformance classes;
- update of the requirements for all conformance services.

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

| FDIS         | Report on voting |
|--------------|------------------|
| 65C/763/FDIS | 65C/773/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.

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

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

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

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

## 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 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.

## **INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –**

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

## **1 Scope**

### **1.1 General**

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

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 14 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible service provided by the Type 14 fieldbus application layer in terms of

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

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

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

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

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

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

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

## 1.2 Specifications

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

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

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

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

## 1.3 Conformance

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

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

## 2 Normative references

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

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

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

IEC 61158-1:2014, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

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

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

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

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

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

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

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

RFC 2030, *Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI*, available at <<http://www.ietf.org>>

ANSI/IEEE 754, *IEEE Standard for Binary Floating-Point Arithmetic*

### **3 Terms, definitions, symbols, abbreviations and conventions**

For the purposes of this document, the following terms, definitions, symbols, abbreviations and conventions as defined in these publications apply:

#### **3.1 ISO/IEC 7498-1 terms**

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

#### **3.2 ISO/IEC 8822 terms**

- a) abstract syntax
- b) presentation context

#### **3.3 ISO/IEC 9545 terms**

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

- i) application control service element

### **3.4 ISO/IEC 8824-1 terms**

- a) object identifier
- b) type

### **3.5 Fieldbus application-layer specific definitions**

#### **3.5.1**

##### **access control**

control on the reading and writing of an object

#### **3.5.2**

##### **access Path**

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

#### **3.5.3**

##### **communication macrocycle**

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

#### **3.5.4**

##### **communication scheduling**

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

#### **3.5.5**

##### **configuration (of a system or device)**

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

#### **3.5.6**

##### **cyclic**

repetitive in a regular manner

#### **3.5.7**

##### **destination FB Instance**

FB instance that receives the specified parameters

#### **3.5.8**

##### **domain**

part of memory used to store code or data

#### **3.5.9**

##### **domain download**

operation to write data in a domain

#### **3.5.10**

##### **domain upload**

operation to read data from a domain

#### **3.5.11**

##### **entity**

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

**3.5.12****bridge**

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

**3.5.13****identifier**

16-bit word associated with a system variable

**3.5.14****index**

address of an object within an application process

**3.5.15****instance**

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

**3.5.16****instantiation**

creation of an instance of a specified type

**3.5.17****management information**

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

**3.5.18****management information base**

organized list of management information

**3.5.19****mapping**

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

**3.5.20****message filtering**

decision on a message according to a special rule

**3.5.21****micro segment**

part of a network, where special scheduling is implemented

**3.5.22****offset**

number of octets from a specially designated position

**3.5.23****phase**

elapsed fraction of a cycle, measured from some fixed origin

**3.5.24****process interface**

data exchange and information mapping between physical process and application unit

**3.5.25****real-time**

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

### 3.5.26

#### **real-time communication**

transfer of data in real-time

### 3.5.27

#### **real-time Ethernet**

#### **RTE**

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

Note 1 to entry: Other communication can be supported, providing the real-time communication is not compromised.

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

### 3.5.28

#### **schedule**

temporal arrangement of a number of related operations

### 3.5.29

#### **scheduling macrocycle**

time interval to implement a specific schedule

### 3.5.30

#### **source FB Instance**

FB instance that sends a specific parameter

### 3.5.31

#### **time offset**

time difference from a specially designated time

## 3.6 Abbreviations and symbols

|         |                                                                 |
|---------|-----------------------------------------------------------------|
| AAE     | Application Access Entity                                       |
| AE      | Application Entity                                              |
| AL      | Application Layer                                               |
| ALME    | Application Layer Management Entity                             |
| ALP     | Application Layer Protocol                                      |
| APO     | Application Object                                              |
| AP      | Application Process                                             |
| APDU    | Application Protocol Data Unit                                  |
| API     | Application Process Identifier                                  |
| AR      | Application Relationship                                        |
| ARP     | Address Resolution Protocol                                     |
| AREP    | Application Relationship End Point                              |
| ASE     | Application Service Element                                     |
| Cnf     | Confirmation                                                    |
| CR      | Communication Relationship                                      |
| CREP    | Communication Relationship End Point                            |
| CSMA/CD | Carrier Sense Multiple Access Protocol with Collision Detection |
| DD      | Device Description                                              |
| DHCP    | Dynamic Host Configuration Protocol                             |
| DL-     | (as a prefix) Data Link-                                        |
| DLCEP   | Data Link Connection End Point                                  |

|             |                                                    |
|-------------|----------------------------------------------------|
| DLL         | Data Link Layer                                    |
| DLE         | Data Link Entity                                   |
| DLM         | Data Link-management                               |
| DLS         | Data Link Service                                  |
| DLSAP       | Data Link Service Access Point                     |
| DLSDU       | DL-service-data-unit                               |
| ECSME       | Type 14 communication scheduling management entity |
| Type 14     | Ethernet for Plant Automation                      |
| EM_         | (as a prefix) Type 14 Management                   |
| ESME        | Type 14 Socket Mapping Entity                      |
| FB          | Function Block                                     |
| FBAP        | Function Block Application Process                 |
| FRT         | Fast Real-time                                     |
| Ind         | Indication                                         |
| IP          | Internet Protocol                                  |
| LLC         | Logical Link Control                               |
| LMP         | Link Management Protocol                           |
| MAC         | Medium Access Control                              |
| MAU         | Medium Attachment Unit                             |
| MOB         | Management Object Base                             |
| PAD         | Pad (bits)                                         |
| PDU         | Protocol Data Unit                                 |
| P/S         | Publisher/Subscriber                               |
| Req         | Request                                            |
| Rsp         | Response                                           |
| RTE         | Real-Time Ethernet                                 |
| RT-Ethernet | Real-Time Ethernet                                 |
| SAP         | Service Access Point                               |
| SDU         | Service Data Unit                                  |
| SME         | System Management Entity                           |
| SNTP        | Simple Network Time Protocol                       |
| TCP         | Transmission Control Protocol                      |
| UDP         | User Datagram Protocol                             |

### 3.7 Conventions

#### 3.7.1 Overview

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

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

#### 3.7.2 Conventions for class definitions

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

| <b>FAL ASE:</b>      |                          | <b>ASE Name</b>        |
|----------------------|--------------------------|------------------------|
| <b>CLASS:</b>        | <b>Class name</b>        |                        |
| <b>CLASS ID:</b>     | <b>#</b>                 |                        |
| <b>PARENT CLASS:</b> | <b>Parent class name</b> |                        |
| <b>ATTRIBUTES:</b>   |                          |                        |
| 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) |
| <b>SERVICES:</b>     |                          |                        |
| 1                    | (o) OpsService:          | service name           |
| 2.                   | (c) Constraint:          | constraint expression  |
| 2.1                  | (o) OpsService:          | service name           |
| 3                    | (m) MgtService:          | service name           |

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

- (e) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
- 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.
  - 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.
  - 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
    - 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).
- (f) The "SERVICES" label indicates that the following entries are services defined for the class.
- 1) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.
  - 2) The label "OpsService" designates an operational service (1).
  - 3) The label "MgtService" designates a management service (2).
  - 4) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

### 3.7.3 Conventions for service definitions

#### 3.7.3.1 General

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

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

#### 3.7.3.2 Service parameters

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

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

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

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

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

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

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

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

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

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

- M parameter is mandatory for the primitive
- U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.
- C parameter is conditional upon other parameters or upon the environment of the service user.
- (blank) parameter is never present.
- S parameter is a selected item.

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

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

### 3.7.3.3 Service procedures

The procedures are defined in terms of

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

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

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

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

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

## 4 Concepts

The common concepts and templates used to describe the application layer service in this standard are detailed in of IEC 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 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 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 an 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 an 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 an 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 1<sup>st</sup>, 1972), the start of the Coordinated Universal Time (UTC) era, until 2151.06.06 (June 6<sup>th</sup>, 2151), i.e. a total range of 65 536 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<sup>st</sup>, 1984.

### 5.3.1.3.7 TimeOfDay with date indication

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

### 5.3.1.3.8 TimeOfDay without date indication

**CLASS:** Data type

**ATTRIBUTES:**

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

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

### 5.3.1.3.9 TIME\_OF\_DAY

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

### 5.3.1.3.10 PrecisionTimeDifference

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.3.11 PrecisionTimeDifference with second indication****CLASS:** Data type**ATTRIBUTES:**

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

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

**5.3.1.3.12 PrecisionTimeDifference without second indication****CLASS:** Data type**ATTRIBUTES:**

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

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

**5.3.1.3.13 TimeValue****CLASS:** Data type**ATTRIBUTES:**

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

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

**5.3.1.3.14 UniversalTime****CLASS:** Data type**ATTRIBUTES:**

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

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

**5.3.1.3.15 FieldbusTime****CLASS:** Data type**ATTRIBUTES:**

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

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

### 5.3.1.4 Numeric types

#### 5.3.1.4.1 BCD

**CLASS:** Data type

**ATTRIBUTES:**

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

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

#### 5.3.1.4.2 Floating Point types

##### 5.3.1.4.2.1 Float32

**CLASS:** Data type

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 8            |
| 2   | Data type Name               | = | Float32      |
| 4   | Format                       | = | FIXED LENGTH |
| 4.1 | Octet Length                 | = | 4            |

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

##### 5.3.1.4.2.2 Floating point

This data type is the same as Float32.

##### 5.3.1.4.2.3 REAL

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

##### 5.3.1.4.2.4 float

This data type is the same as Float32.

##### 5.3.1.4.2.5 Float64

**CLASS:** Data type

**ATTRIBUTES:**

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

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

##### 5.3.1.4.2.6 LREAL

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

**5.3.1.4.2.7 double**

This data type is the same as Float64.

**5.3.1.4.3 Integer types****5.3.1.4.3.1 Integer8**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.4.3.2 SINT**

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

**5.3.1.4.3.3 char**

This data type is the same as Integer8.

**5.3.1.4.3.4 Integer16**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.4.3.5 INT**

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

**5.3.1.4.3.6 short**

This data type is the same as Integer16.

**5.3.1.4.3.7 Integer32**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.4.3.8 DINT**

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

#### 5.3.1.4.3.9 long

This data type is the same as Integer32.

#### 5.3.1.4.3.10 Integer64

**CLASS:** Data type

**ATTRIBUTES:**

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

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

#### 5.3.1.4.3.11 LINT

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

#### 5.3.1.4.4 Unsigned types

##### 5.3.1.4.4.1 Unsigned8

**CLASS:** Data type

**ATTRIBUTES:**

|     |                              |   |              |
|-----|------------------------------|---|--------------|
| 1   | Data type Numeric Identifier | = | 5            |
| 2   | Data type Name               | = | Unsigned8    |
| 3   | Format                       | = | FIXED LENGTH |
| 4.1 | Octet Length                 | = | 1            |

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

##### 5.3.1.4.4.2 USINT

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

##### 5.3.1.4.4.3 unsigned char

This data type is the same as Unsigned8.

##### 5.3.1.4.4.4 Unsigned16

**CLASS:** Data type

**ATTRIBUTES:**

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

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

##### 5.3.1.4.4.5 UINT

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

**5.3.1.4.4.6 unsigned short**

This data type is the same as Unsigned16.

**5.3.1.4.4.7 Unsigned32**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.4.4.8 UDINT**

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

**5.3.1.4.4.9 unsigned long**

This data type is the same as Unsigned32.

**5.3.1.4.4.10 Unsigned64**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.4.4.11 ULINT**

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

**5.3.1.4.4.12 Normalised 2 octet**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

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

SN = 0: positive numbers including zero

SN = 1: negative numbers.

**Table 1 – Attribute of variable normalised 2 octet**

| Data type | Range of values                        | Resolution            | Length   |
|-----------|----------------------------------------|-----------------------|----------|
| N2        | $-200 \% \leq i \leq (200-2^{-14}) \%$ | $2^{-14} = 0,0061 \%$ | 2 Octets |

**Table 2 – Encoding of variable normalised 2 octet**

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

#### 5.3.1.4.4.13 Normalised 4 octet

**CLASS:** Data type

**ATTRIBUTES:**

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

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

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

SN = 0: positive numbers including zero

SN = 1: negative numbers.

**Table 3 – Attribute of normalised 4 Octet**

| Data type | Range of values                        | Resolution                   | Length   |
|-----------|----------------------------------------|------------------------------|----------|
| N4        | $-200 \% \leq i \leq (200-2^{-30}) \%$ | $2^{-30} = 9,3 * 10^{-8} \%$ | 4 Octets |

**Table 4 – Encoding of normalised 4 Octet**

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

#### 5.3.1.4.4.14 Variable normalised 2 octet

**CLASS:** Data type

**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 123
- 2 Data type Name = X2

3 Format = FIXED LENGTH

4.1 Octet Length = 2

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

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

SN = 0: positive numbers including zero

SN = 1: negative numbers.

**Table 5 – Attribute of variable normalised 2 octet**

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

**Table 6 – Encoding of variable normalised 2 octet**

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

#### 5.3.1.4.4.15 Variable Normalised 4 Octet

**CLASS:** Data type

**ATTRIBUTES:**

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

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

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

SN = 0: positive numbers including zero

SN = 1: negative numbers.

**Table 7 – Attribute of variable normalised 4 Octet**

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

**Table 8 – Encoding of variable normalised 4 Octet**

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

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

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

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

NOTE 1 These types do not exist in ASN.1, they are expressed in IEC 60870 as 'unsigned fixed point number'.

NOTE 2 These types are defined in IEC 61375.

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

**Table 9 – Attribute of unipolar 2 octet**

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

**Table 10 – Encoding of unipolar 2 octet**

| Octet        | Bit             |       |          |          |          |          |          |          |          |          |          |           |           |           |           |           |
|--------------|-----------------|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|
|              | 0               | 1     | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        | 10       | 11        | 12        | 13        | 14        | 15        |
| 1,2          | $2^1$           | $2^0$ | $2^{-1}$ | $2^{-2}$ | $2^{-3}$ | $2^{-4}$ | $2^{-5}$ | $2^{-6}$ | $2^{-7}$ | $2^{-8}$ | $2^{-9}$ | $2^{-10}$ | $2^{-11}$ | $2^{-12}$ | $2^{-13}$ | $2^{-14}$ |
| Integer part | Fractional part |       |          |          |          |          |          |          |          |          |          |           |           |           |           |           |

**5.3.1.4.4.17 Fixed point value 2 Octet****CLASS:** Data type**ATTRIBUTES:**

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

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

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

SN = 0: positive numbers including zero

SN = 1: negative numbers.

**Table 11 – Attribute of Fixed point value 2 Octet**

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

**Table 12 – Encoding of Fixed point value 2 Octet**

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

#### 5.3.1.4.4.18 Fixed point value 4 Octet

**CLASS:** Data type

**ATTRIBUTES:**

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

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

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

**Table 13 – Attribute of Fixed point value 4 Octet**

| Data type | Range of values                                | Resolution         | Length   |
|-----------|------------------------------------------------|--------------------|----------|
| C4        | $-214\ 748,364\ 8 \leq i \leq 214\ 748,364\ 7$ | $10^{-4} = 0,0001$ | 4 Octets |

#### 5.3.1.4.4.19 Bit sequence 2 Octet

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**Table 14 – Encoding of Bit sequence 2 Octet**

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

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

- 1 Data type Numeric Identifier = 116
- 2 Data type Name = L2
- 3 Format = FIXED LENGTH
- 4.1 Octet Length = 4

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

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

**Table 15 – Encoding of Nibble 4 Octet**

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

**5.3.1.5 OctetString character types****5.3.1.5.1 OctetString1****CLASS:** Data type**ATTRIBUTES:**

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

This type has a length of one octet.

**5.3.1.5.2 OctetString2****CLASS:** Data type**ATTRIBUTES:**

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

This type has a length of two octets.

**5.3.1.5.3 OctetString4****CLASS:** Data type**ATTRIBUTES:**

- 1 Data type Numeric Identifier = 32

|     |                |   |              |
|-----|----------------|---|--------------|
| 2   | Data type Name | = | OctetString4 |
| 3   | Format         | = | FIXED LENGTH |
| 4.1 | Octet Length   | = | 4            |

This type has a length of four octets.

#### 5.3.1.5.4 OctetString8

**CLASS:** Data type

**ATTRIBUTES:**

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

This octet string type has a length of eight octets.

#### 5.3.1.5.5 OctetString16

**CLASS:** Data type

**ATTRIBUTES:**

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

This type has a length of 16 octets.

#### 5.3.1.6 Pointer types

##### 5.3.1.6.1 Interface pointer

**CLASS:** Data type

**ATTRIBUTES:**

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

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

##### 5.3.1.6.2 LPWSTR

**CLASS:** Data type

**ATTRIBUTES:**

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

This data type defines a reference to an UnicodeString.

#### 5.3.1.7 Time types

##### 5.3.1.7.1 BinaryTime0

**CLASS:** Data type

**ATTRIBUTES:**

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

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

#### 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  $\mu$ 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  $\mu$ 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  $\mu$ 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  $\mu$ s.

**5.3.1.7.10 BinaryTime9****CLASS:** Data type**ATTRIBUTES:**

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

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

#### 5.3.1.7.11 TIME

**CLASS:** Data type

**ATTRIBUTES:**

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

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

#### 5.3.1.7.12 ITIME

**CLASS:** Data type

**ATTRIBUTES:**

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

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

#### 5.3.1.7.13 FTIME

**CLASS:** Data type

**ATTRIBUTES:**

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

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

#### 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  $\mu$ s.

#### 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  $0 \leq x \leq 32\,767$ . When interpreted, internal values outside this range are set to 0. The attribute of data type is shown in Table 16.

**Table 16 – Attribute of multiple time constant 2 octets**

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

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

#### 5.3.1.7.18 Multiple time constant 4 octets

**CLASS:** Data type

**ATTRIBUTES:**

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

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

As by Unsigned32 with a restricted Range of values  $0 \leq x \leq 4\,294\,967\,295$ . When interpreted, internal values outside this range are set to 0. The attribute of data type is shown in Table 17.

**Table 17 – Attribute of multiple time constant 4 octets**

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

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

#### 5.3.1.7.19 Fraction time constant 2 octets

**CLASS:** Data type

**ATTRIBUTES:**

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

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

As by Unsigned16 with a restricted Range of values  $0 \leq x \leq 32\,767$ . When interpreted, internal values outside this range are set to 0. Interpreted value = Internal value \*  $T_a/16\,384$ . The attribute of data type is shown in Table 18.

**Table 18 – Attribute of fraction time constant 2 octets**

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

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

#### 5.3.1.7.20 Reciprocal time constant 2 octets

**CLASS:** Data type

**ATTRIBUTES:**

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

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

As by Unsigned16 with a limited Range of values  $1 \times 16\,384$ . When interpreted, internal values outside this range are set to 16 384. Interpreted value =  $16\,384 \times T_a/\text{internal value}$ . The attribute of data type is shown in Table 19.

**Table 19 – Encoding of reciprocal time constant 2 octets**

| Data type | Range of values                               | Resolution | Length  |
|-----------|-----------------------------------------------|------------|---------|
| R2        | $1 \times T_a \leq i \leq 16\,384 \times T_a$ | $T_a$      | 2 Octet |

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

#### 5.3.1.8 VisibleString character types

##### 5.3.1.8.1 UNICODE char

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.8.2 VisibleString1**

**CLASS:** Data type

**ATTRIBUTES:**

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

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

**5.3.1.8.3 VisibleString2**

**CLASS:** Data type

**ATTRIBUTES:**

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

This type contains two elements of type VisibleString.

**5.3.1.8.4 VisibleString4**

**CLASS:** Data type

**ATTRIBUTES:**

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

This type contains four elements of type VisibleString.

**5.3.1.8.5 VisibleString8**

**CLASS:** Data type

**ATTRIBUTES:**

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

This type contains eight elements of type VisibleString.

**5.3.1.8.6 VisibleString16**

**CLASS:** Data type

**ATTRIBUTES:**

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

This type contains 16 elements of type VisibleString.

## 5.3.2 String types

### 5.3.2.1 BitString

**CLASS:** Data type

**ATTRIBUTES:**

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

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

### 5.3.2.2 CompactBooleanArray

**CLASS:** Data type

**ATTRIBUTES:**

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

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

### 5.3.2.3 CompactBCDArray

**CLASS:** Data type

**ATTRIBUTES:**

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

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

### 5.3.2.4 OctetString

**CLASS:** Data type

**ATTRIBUTES:**

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

An OctetString is an ordered sequence of octets, numbered from 1 to n. For the purposes of discussion, octet 1 of the sequence is referred to as the first octet. IEC 61158-6-14 defines the order of transmission.

### 5.3.2.5 UNICODEString

**CLASS:** Data type

**ATTRIBUTES:**

|   |                              |   |               |
|---|------------------------------|---|---------------|
| 1 | Data type Numeric Identifier | = | 39            |
| 2 | Data type Name               | = | UnicodeString |

3        Format                                =    STRING

4.1     Octet Length                        =    2

This type is defined as the UNICODE string type.

### 5.3.2.6 VisibleString

**CLASS:**                                        Data type

**ATTRIBUTES:**

1        Data type Numeric Identifier    =    9

2        Data type Name                    =    VisibleString

3        Format                                =    STRING

4.1     Octet Length                        =    1 to n

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

## 5.4 Data type ASE service specification

There are no operational services defined for the type object.

## 6 Communication model specification

### 6.1 General

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

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

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

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

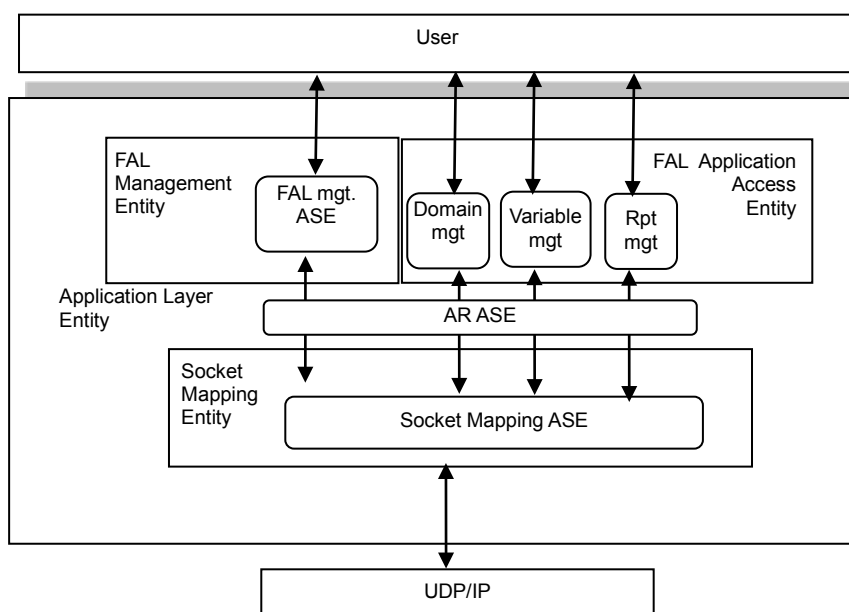


Figure 1 – Application layer entity

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

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

## **6.2 ASEs**

### **6.2.1 FAL management ASE**

#### **6.2.1.1 Overview**

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

##### **6.2.1.1.1 Device identification**

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

#### **Device ID**

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

#### **PD\_Tag**

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

#### **Redundancy number**

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

##### **6.2.1.1.2 Address assignment**

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

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

##### **6.2.1.1.3 Object location**

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

FAL Management entity provides EM\_DetectingDevice service to query the IP address of a device using PD\_Tag. It also provides EM\_GetDeivceAttibute 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 RFC 2030 or IEC 61588 can be used according to practical application.

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

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

#### **6.2.1.1.6 Link management**

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

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

#### **6.2.1.1.7 Management object base**

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

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

**Table 20 – Management object base**

| Object                            | Object ID                | Illustration                                  |
|-----------------------------------|--------------------------|-----------------------------------------------|
| MIB Header                        | 1                        | device Management Object Base header object   |
| Device Descriptor                 | 2                        | device descriptor object                      |
| Time Synchronization              | 3                        | Time Synchronization Object                   |
| Max Response Time                 | 4                        | Confirmed Service Max Response Time           |
| Communication Schedule Management | 5                        | Communication Schedule Management Object      |
| Device Application information    | 6                        | Device Application information Object         |
| FB Application information Header | 7                        | Function Block Application information Header |
| Link Object Header                | 8                        | Link Object Header                            |
| Domain Application Object Header  | 9                        | Domain Application Object Header              |
| FRT Link Object Header            | 12                       | FRT Link Object Header                        |
| .....                             |                          |                                               |
| FB Application information 1      | 2000                     | Function Block Application information 1      |
| FB Application information 2      | 2001                     | Function Block Application information 2      |
| .....                             | Increased number in turn |                                               |
| Domain Application Object 1       | 4000                     | Domain Application Object 1                   |
| Domain Application Object 2       | 4001                     | Domain Application Object 2                   |
| .....                             | Increased number in turn |                                               |
| Link Object 1                     | 5000                     | Link Object 1                                 |
| Link Object 2                     | 5001                     | Link Object 2                                 |
| .....                             | Increased number in turn |                                               |
| FRT Link Object 1                 | 7000                     | FRT Link Object 1                             |
| FRT Link Object 2                 | 7001                     | FRT Link Object 2                             |
| .....                             | Increased number in turn |                                               |

## 6.2.1.2 FAL management objects

### 6.2.1.2.1 MOB header class

MOB Header Class represents the version number of MOB.

#### 6.2.1.2.1.1 Formal model

MOB header class is presented as follows:

|                       |                           |
|-----------------------|---------------------------|
| <b>ASE:</b>           | <b>FAL MANAGEMENT ASE</b> |
| <b>CLASS:</b>         | <b>MOB HEADER</b>         |
| <b>CLASS ID:</b>      | <b>Not Used</b>           |
| <b>PARENT CLASS:</b>  | <b>TOP</b>                |
| <b>ATTRIBUTES:</b>    |                           |
| 1. (m) Key Attribute: | Object ID                 |
| 2. (m) Attribute:     | MOB Revision Number       |
| <b>SERVICES:</b>      |                           |
| 1. (o) OpService:     | 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

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

#### 6.2.1.2.2.2 Attributes

#### Object ID

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

#### Reserved

This field is reserved.

**Application type**

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

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

**Device ID**

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

**PD\_Tag**

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

**Active IP address**

This attribute indicates current operable IP address.

**Device type**

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

**Status**

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

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

**Device version**

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

**Annunciation interval**

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

**Annunciation version number**

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

**Device redundancy state**

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

**Device redundancy number**

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

**Max redundancy number**

This attribute specifies the max redundancy number.

**LAN redundancy port**

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

**Duplicate tag detected**

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

**6.2.1.2.2.3 Services****EM\_DetectingDevice**

This optional service allow user to find device information according to device PD\_Tag.

**EM\_OnlineReply**

This optional service is used to reply the device which sends EM\_DetectingDevice request.

**EM\_ConfiguringDevice**

This optional service allows user to set the device attributes.

**EM\_GetDeviceAttribute**

This optional service allows user to get the device attributes.

**EM\_SetDefaultValue**

This optional service allows user to clear the device attributes.

**EM\_ActiveNotification**

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

**6.2.1.2.3 Time synchronization class**

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

**6.2.1.2.3.1 Formal model**

This class is presented as follows:

|                      |                                     |
|----------------------|-------------------------------------|
| <b>ASE:</b>          | <b>FAL MANAGEMENT ASE</b>           |
| <b>CLASS:</b>        | <b>TIME SYNCHRONIZATION</b>         |
| <b>CLASS ID:</b>     | <b>Not Used</b>                     |
| <b>PARENT CLASS:</b> | <b>TOP</b>                          |
| <b>ATTRIBUTES:</b>   |                                     |
| 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
- 8: Time Synchronization precision < 100 ns
- 9: Time Synchronization precision < 10 ns

**Target time sync class**

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

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

- 5: Time Synchronization precision <100 µs;
- 6: Time Synchronization precision <10 µs;
- 7: Time Synchronization precision <1 µs
- 8: Time Synchronization precision < 100 ns
- 9: Time Synchronization precision < 10 ns

### Current time

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

### Standard time difference

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

### 6.2.1.2.3.3 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

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

#### 6.2.1.2.4.2 Attributes

#### Object ID

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

#### Reserved

This field is reserved.

#### Max response time

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

#### 6.2.1.2.4.3 Services

##### Read

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

##### Write

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

#### 6.2.1.2.5 Communication scheduling management class

This class specifies the information used in an ECSME.

##### 6.2.1.2.5.1 Formal model

|                      |                                                    |
|----------------------|----------------------------------------------------|
| <b>ASE:</b>          | <b>FAL MANAGEMENT ASE</b>                          |
| <b>CLASS:</b>        | <b>COMMUNICATION SCHEDULE MANAGEMENT</b>           |
| <b>CLASS ID:</b>     | <b>Not Used</b>                                    |
| <b>PARENT CLASS:</b> | <b>TOP</b>                                         |
| <b>ATTRIBUTES:</b>   |                                                    |
| 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 |
| <b>SERVICES:</b>     |                                                    |
| 1. (o)               | OpsService: Read                                   |
| 2. (o)               | OpsService: Write                                  |

##### 6.2.1.2.5.2 Attributes

##### Object ID

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

##### Communication macrocycle

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

##### Non-periodic data transfer offset

This attribute specifies time offset of the NonPeriodicDataTransfer starting time from the starting time of a communication macrocycle. Its data type is 8 octets of PrecisionTimeDifference. The default value of 0XFFFF FFFF FFFF FFFF 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.