

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 5-20: Application layer service definition – Type 20 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-20: Définition des services de la couche application – Éléments
de type 20**

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014



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
www.iec.ch

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

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

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

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

IEC Customer Service Centre - webstore.iec.ch/csc

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

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

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

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

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

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

Service Clients - webstore.iec.ch/csc

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



IEC 61158-5-20

Edition 3.0 2014-08

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 5-20: Application layer service definition – Type 20 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 5-20: Définition des services de la couche application – Éléments
de type 20**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XA

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-1741-2

**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.....	3
INTRODUCTION.....	5
1 Scope.....	6
2 Normative references	6
3 Terms, definitions, symbols, abbreviations and conventions	7
3.1 Terms and definitions from other ISO/IEC standards	7
3.2 IEC 61158-1 terms	8
3.3 Type 20 fieldbus application-layer specific definitions	10
3.4 Abbreviations and symbols	12
3.5 Conventions	13
4 Concepts	16
5 Data type ASE	16
5.1 Overview	16
5.2 Formal definition of data type objects	18
5.3 FAL defined data types.....	20
5.4 Data type ASE service specification	23
5.5 Summary of data types	24
6 Communication model specification	24
6.1 Common parameters	24
6.2 ASEs	25
6.3 ARs	52
6.4 Summary of classes	54
6.5 Permitted services by AREP role.....	55
Bibliography.....	56
Figure 1 – Data type class hierarchy	17
Figure 2 – VFD model	25
Table 1 – Packed ASCII character set.....	23
Table 2 – ISO Latin-1 characters	23
Table 3 – Data type summary	24
Table 4 – Response code values	24
Table 5 – Communication status values	25
Table 6 – Identify service parameters	28
Table 7 – Read service parameters.....	32
Table 8 – Write service parameters.....	33
Table 9 – Information report parameters	34
Table 10 – Action service parameters	36
Table 11 – AR get attributes service parameters	53
Table 12 – AR set attributes service parameters	54
Table 13 – Class summary.....	55
Table 14 – Confirmed services by AREP class	55
Table 15 – Unconfirmed services by AREP class	55

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –****Part 5-20: Application layer service definition –
Type 20 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 the associated protocol type is restricted by its 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 its intellectual-property-right holders.

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

International Standard IEC 61158-5-20 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 change with respect to the previous edition is listed below:

- added Data types;
- added services;
- updated the Normative references, Terms, definitions, symbols, abbreviations;
- corrected the editorial errors and the text.

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.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014

INTRODUCTION

This document is one of a series produced to facilitate the interconnection of automation system components. It is related to other documents 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 document 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 document is a conceptual architectural service, independent of administrative and implementation divisions.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-20: Application layer service definition –

1 Scope

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 International 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 20 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 International Standard defines in an abstract way the externally visible service provided by the Type 20 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 International Standard is to define the services provided to the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model.

This International Standard specifies the structure and services of the IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

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 International Standard to provide access to the FAL to control certain aspects of its operation.

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

IEC 62591:2010, *Industrial communication networks – Wireless communication network and communication profiles – WirelessHART™*

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

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

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

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*

ANSI/IEEE 754: *IEEE Standard for Floating-Point Arithmetic*

3 Terms, definitions, symbols, abbreviations and conventions

For the purposes of this document, the following terms, definitions, abbreviations, symbols and conventions apply.

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 ISO/IEC 7498-1 terms

- a) abstract syntax
- b) application entity
- c) application process
- d) application protocol data unit
- e) application service element

3.1.2 ISO/IEC 9545 terms

- a) application-entity-invocation
- b) application-service-element
- c) application-service-element

3.1.3 ISO/IEC 8824-1 terms

- a) object identifier
- b) type
- c) value
- d) simple type
- e) structured type
- f) component type

- g) tag
- h) true
- i) false
- j) integer type
- k) octet string type
- m) null type

3.2 IEC 61158-1 terms

For the purposes of this document, the following terms and definitions apply.

3.2.1

application

function or data structure for which data is consumed or produced

3.2.2

application object

object class that manages and provides the run time exchange of messages across the network and within the network device

Note 1 to entry: Multiple types of application object classes may be defined.

3.2.3

application process

part of a distributed application on a network, which is located on one device and unambiguously addressed

3.2.4

application process object

component of an application process that is identifiable and accessible through an FAL application relationship

Note 1 to entry: Application process object definitions are composed of a set of values for the attributes of their class (see the definition for Application Process Object Class Definition). Application process object definitions may be accessed remotely using the services of the FAL Object Management ASE. FAL Object Management services can be used to load or update object definitions, to read object definitions, and to dynamically create and delete application objects and their corresponding definitions

3.2.5

application process object class

class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.6

application relationship

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation

Note 1 to entry: This relationship is activated either by the exchange of application-protocol-data-units or as a result of pre-configuration activities

3.2.7

application relationship endpoint

context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

Note 1 to entry: Each application process involved in the application relationship maintains its own application relationship endpoint

3.2.8**attribute**

description of an externally visible characteristic or feature of an object

Note 1 to entry: The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behaviour of an object. Attributes are divided into class attributes and instance attributes

3.2.9**behaviour**

indication of how the object responds to particular events

Note 1 to entry: Its description includes the relationship between attribute values and services.

3.2.10**class**

set of objects, all of which represent the same kind of system component

Note 1 to entry: A class is a generalisation of the object; a template for defining variables and methods. All objects in a class are identical in form and behaviour, but usually contain different data in their attributes

3.2.11**class attributes**

attribute that is shared by all objects within the same class

3.2.12**class code**

unique identifier assigned to each object class

3.2.13**class specific service**

service defined by a particular object class to perform a required function which is not performed by a common service

Note 1 to entry: A class specific object is unique to the object class which defines it.

3.2.14**client**

- a) an object which uses the services of another (server) object to perform a task
- b) an initiator of a message to which a server reacts, such as the role of an AR endpoint in which it issues confirmed service request APDUs to a single AR endpoint acting as a server

3.2.15**conveyance path**

unidirectional flow of APDUs across an application relationship

3.2.16**cyclic**

term used to describe events which repeat in a regular and repetitive manner

3.2.17**endpoint**

one of the communicating entities involved in a connection

3.2.18**error**

discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.2.19

error code

identification of a specific type of error within an error class

3.2.20

management information

network-accessible information that supports managing the operation of the fieldbus system, including the application layer

Note 1 to entry: Managing includes functions such as controlling, monitoring, and diagnosing.

3.2.21

server

- a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request
- b) an object which provides services to another (client) object

3.2.22

service

operation or function than an object and/or object class performs upon request from another object and/or object class

Note 1 to entry: A set of common services is defined and provisions for the definition of object-specific services are provided. Object-specific services are those which are defined by a particular object class to perform a required function which is not performed by a common service.

3.3 Type 20 fieldbus application-layer specific definitions

For the purposes of this document, the following terms and definitions apply.

3.3.1

analog channel

continuously varying electrical signal connecting a field device to the remainder of the data acquisition or control system

Note 1 to entry: Some field devices support multiple analog channels (input or output). Each analog channel transmits a single dynamic variable to or from the field device.

3.3.2

broadcast

process of sending a PDU to all devices that are connected to the network and are able to receive the transmission

3.3.3

broadcast address

address used by a master to send a command to all devices

3.3.4

burst mode

initiation of communication activity by a slave device at cyclic interval without request from a master

3.3.5

comm error

detectable error in receiving a PhPDU or DLPDU, also 'Communication error code' octet of APDU

3.3.6**device**

any entity containing an implementation of Type 20 fieldbus

3.3.7**device ID**

serial number for a device that is unique among all instances of one type of device

Note 1 to entry: The manufacturer is required to assigned unique value for every device that has the identical values for Manufacturer ID and Device Type.

3.3.8**device type**

manufacturer's type of a device, e.g. its product name

Note 1 to entry: The value of this attribute is unique among all manufacturers and all type of devices. Its value specifies the set of commands and data objects supported by the device.

3.3.9**device variable**

uniquely defined data item within a Field Device that is always associated with the process-related information

Note 1 to entry: A device variable's value varies in response to changes and variations in the process to which the device is connected.

3.3.10**dynamic variable**

device variable that is assigned as the dynamic variable and possibly associated with an analog channel

Note 1 to entry: A device may contain up to four variables – primary, secondary, tertiary, and quaternary variables. These are collectively called the dynamic variables.

3.3.11**expanded device type**

manufacturer's type of the device

Note 1 to entry: The value of this attribute is unique among all manufacturers and all type of devices. Its value specifies the set of commands and data objects supported by the device.

3.3.12**field device**

physical entity that is connected to the process or to plant equipment and has at least one signalling element that communicates with other signalling element(s) via the network

Note 1 to entry: It directly connects to the sensor or actuator or performs process control function and it is directly connected to the physical layer specified in this standard. It may generate or receive an analog signal in addition to a digital signal.

3.3.13**long tag**

32 character restricted ISO Latin-1 string used to identify a field device

3.3.14**loop current**

value measured by a milli-ammeter in series with the field device

Note 1 to entry: The loop current is a near DC analog 4-20 mA signal used to communicate a single value between the control system and the field device. Voltage mode devices use "Volts DC" as their engineering units where "loop current" values are used.

3.3.15

manufacturer ID

string identifying the manufacturer that produced the device

Note 1 to entry: A manufacturer is required to use the value assigned to it and is not permitted to use the value assigned to another manufacturer.

3.3.16

master

device that initiates communication activity by sending request PDU to a another device and expecting a response frame from that device

3.3.17

network

single pair of cable, connectors, associated signaling elements by which a given set of signaling devices are interconnected and non-signaling elements that are attached to the same pair of cable

Note 1 to entry: An installation using multiple-pair wire and a common network power supply is considered as multiple networks.

3.3.18

polling address

identifier assigned to a device such that it is unique within the network to which the device is connected

3.3.19

slave

device that initiates communication activity only after it receives a request PDU from a master device and it is required to send a response to that request

3.3.20

tag

8 character ASCII string used to identify the field device

3.3.21

unique ID

identifier assigned to a device which is unique among all instances of the devices compliant to this standard

3.4 Abbreviations and symbols

AE	Application Entity
AL	Application Layer
AP	Application Process
APDU	Application Protocol Data Unit
APO	Application Process Object
AR	Application Relationship
AREP	Application Relationship End Point
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
Cnf	Confirmation
DL-	(as a prefix) Data Link-
DLC	Data Link Connection
DLL	Data Link Layer

DLM	Data Link-management
DLSAP	Data Link Service Access Point
DLSDU	DL-service-data-unit
FAL	Fieldbus Application Layer
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
OSI	Open Systems Interconnect
Req	Request
Rsp	Response
VFD	Virtual Field Device

3.5 Conventions

3.5.1 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 service specification defines the services that are provided by the ASE.

3.5.2 Conventions for class definitions

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

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

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

- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. The CLASS ID is not used in this document.
- (4) The "PARENT CLASS:" entry is the name of the parent class for the class being specified. All attributes defined for the parent class and inherited by it are inherited for the class being defined, and therefore do not have to be redefined in the template for this class.

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

- (5) The "ATTRIBUTES" label indicate that the following entries are attributes defined for the class.
 - a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
 - b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
 - c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify
 - i) fields of a structured attribute (4.1, 4.2, 4.3),
 - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).
 - iii) the selection fields of a choice type attribute (6.1 and 6.2).
- (6) The "SERVICES" label indicates that the following entries are services defined for the class.
 - a) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.
 - b) The label "OpsService" designates an operational service.
 - c) The label "MgtService" designates an management service.
 - d) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.5.3 Conventions for service definitions

3.5.3.1 General

This document 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.

3.5.3.2 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.

This document 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 document. Each table consists of up to six columns: a column for the name of the service parameter, and a column each for those primitives and parameter-transfer directions used by the service. The possible six columns are

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

NOTE 1 This is a seldom-used capability. Unless otherwise specified, request primitive parameters are input parameters.

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

NOTE 2 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).

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

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

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

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

3.5.3.3 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 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 subseries protocol specification document but not in an IEC 61158-5 subseries abstract service definition document. Similarly, local implementation methods by which a service provider or service user pairs request and confirm(ation) primitives, or indication and response primitives, is appropriate for an implementation specification or for a programming interface specification, but not for an abstract service document or for a protocol document, except at a level of abstraction that is necessarily common to all embodiments of the specifying document. 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 document are detailed in IEC 61158-1, Clause 9.

5 Data type ASE

5.1 Overview

5.1.1 General

Fieldbus data types specify the machine independent syntax for application data conveyed by FAL services. The Fieldbus application layer supports the definition and transfer of both basic and constructed data types. Encoding rules for the data types specified in this clause are provided in IEC 61158-6 subseries.

Basic types are atomic types that cannot be decomposed into more elemental types. Constructed types are types composed of basic types and other constructed types. Their complexity and depth of nesting is not constrained by this document.

Data types are defined as instances of the data type class, as shown in Figure 1.

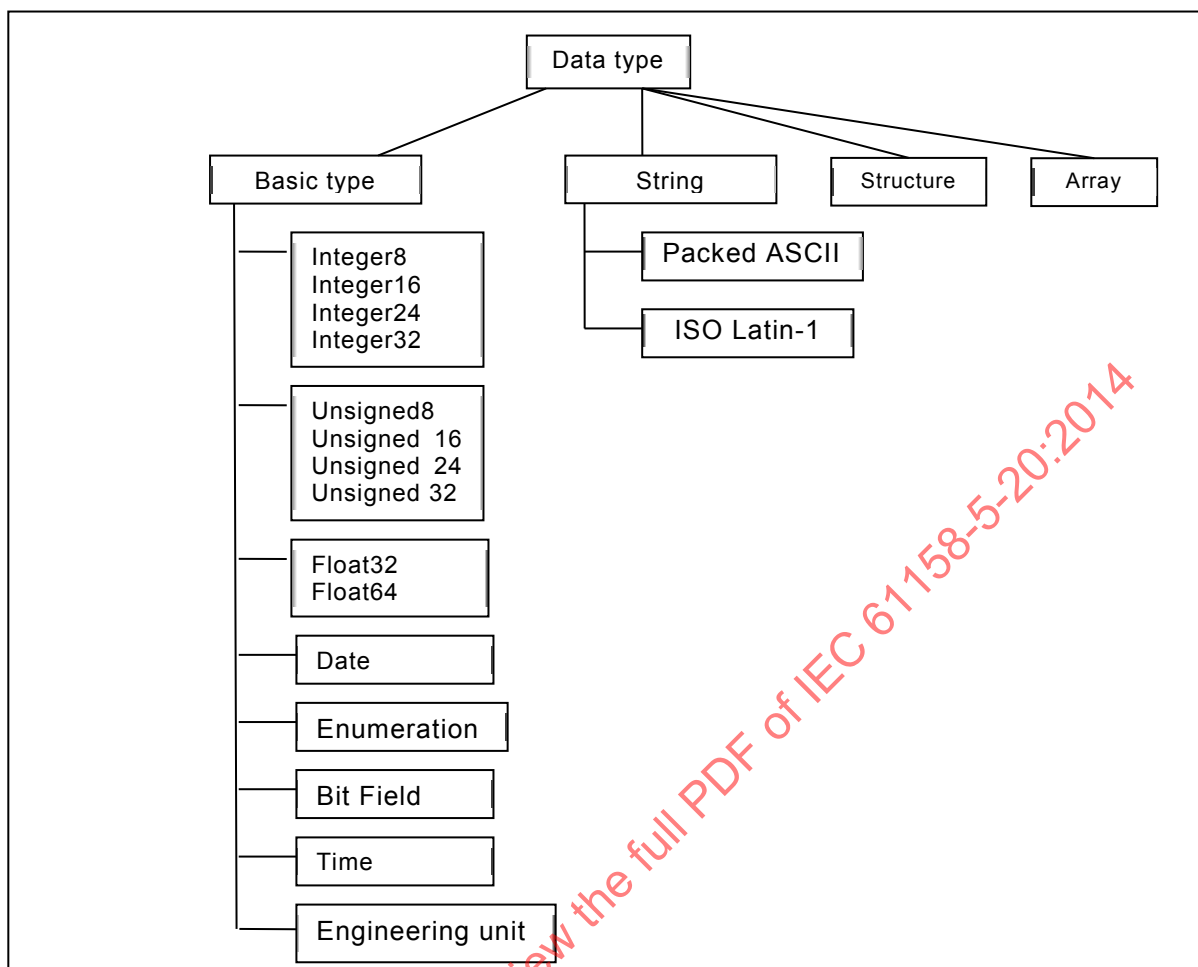


Figure 1 – Data type class hierarchy

The data type definitions are represented as a class/format/instance structure beginning with data type class entitled "Data type". The formats for data types are defined by the data type class.

The basic data classes are always fixed length data types. Standard types taken from ISO/IEC 8824-1 are referred to as *simple* data types. Other standard basic data types are defined specifically for Fieldbus applications and are referred to as *specific types*.

The constructed types specified in this document are strings, arrays and structures. There are no standard types defined for arrays and structures.

5.1.2 Basic types

Most basic types are defined from a set of ISO/IEC 8824-1 types (simple types). Some ISO/IEC 8824-1 types have been extended for Fieldbus specific use (specific types).

Simple types are ISO/IEC 8824-1 universal types. They are defined in this document to provide them with Fieldbus class identifiers.

Specific types are basic types defined specifically for use in the Fieldbus environment. They are defined as simple class subtypes.

Basic types have a constant length. Two variations are defined, one for defining data types whose length is an integral number of octets, and one for defining data types whose length is bits.

NOTE Integer, Packed ASCII, ISO Latin-1 and Date are defined in this document for the purpose of assigning Fieldbus class identifiers to them. This document does not change their definitions as specified in ISO/IEC 8824-1.

5.1.3 Constructed types

5.1.3.1 Overview

Constructed data types are needed to completely convey the variety of information present on the Fieldbus. There are three kinds of constructed types defined for this document – string, array and structure.

5.1.3.2 String

A string is composed of an ordered set, variable in number, of homogeneously typed fixed-length elements.

5.1.3.3 Structure

A structure is made of an ordered set of heterogeneously typed elements called fields. This document does not restrict the data type of fields. However, the fields within a structure do not have to be of the same type.

5.1.3.4 Array

An array is composed of an ordered set of homogeneously typed elements. The data type of array elements can be fixed length basic type or structure. All elements of an array shall be of the same type.

5.1.3.5 Nesting level

This document permits structures and arrays to contain structures and arrays.

5.1.4 Specification of user defined data types

Users may find it necessary to define custom data types for their own applications. User defined types are not supported by this document.

5.1.5 Transfer of user data

User data is transferred between applications by the FAL protocol. All encoding and decoding are performed by the FAL user.

The rules for encoding user data in FAL protocol data units is data type dependent. These rules are defined in IEC 61158-6-20. User-defined data types for which there are no encoding rules are transferred as a variable-length sequence of octets. The format of the data within the octet string is defined by the user.

5.2 Formal definition of data type objects

5.2.1 Data type class

The data type class specifies the root of the data type class tree. Its parent class "top" indicates the top of the FAL class tree.

FAL ASE: DATA TYPE ASE
CLASS: DATA TYPE

CLASS ID: Not used

PARENT CLASS: TOP

ATTRIBUTES:

1	(m)	Key Attribute:	Data type Name
2	(m)	Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
3	(c)	Constraint:	Format = FIXED LENGTH STRING
3.1	(m)	Attribute:	Octet Length
4	(c)	Constraint:	Format = STRUCTURE
4.1	(m)	Attribute:	Number of Fields
4.2	(m)	Attribute:	List of Fields
4.2.1	(o)	Attribute:	Field Name
4.2.2	(m)	Attribute:	Field Data type
5	(c)	Constraint:	Format = ARRAY
5.1	(m)	Attribute:	Number of Array Elements
5.2	(m)	Attribute :	Array Element Data type

5.2.2 Attributes

Format

This attribute identifies the data type as a fixed-length, string, array, or data structure.

Octet Length

This conditional attribute defines the representation of the dimensions of the associated type object. It is present when the value of the format attribute is "FIXED LENGTH" or "STRING". For FIXED LENGTH data types, it represents the length in octets. For STRING data types, it represents the length in octets for a single element of a string.

Number of Fields

This conditional attribute defines the number of fields in a structure. It is present when the value of the format attribute is "STRUCTURE".

List of Fields

This conditional attribute is an ordered list of fields contained in the structure. Each field is specified by its number and its type. Fields are numbered sequentially from 0 (zero) in the order in which they occur. Partial access to fields within a structure is not supported.

Field Name

This conditional, optional attribute specifies the name of the field. It may be present when the value of the format attribute is "STRUCTURE".

Field Data type

This conditional attribute specifies the data type of the field. It is present when the value of the format attribute is "STRUCTURE". This attribute may itself specify a constructed data type by referencing a constructed data type definition by embedding a constructed data type definition here.

Number of Array Elements

This conditional attribute defines the number of elements for the array type. Array elements are indexed starting at "0" through "n-1" where the size of the array is "n" elements. This attribute is present when the value of the format attribute is "ARRAY".

Array Element Data type

This conditional attribute specifies the data type for the elements of an array. All elements of the array have the same data type. It is present when the value of the format attribute is "ARRAY". This attribute may itself specify a constructed data type by referencing a constructed data type by its name.

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Integer8

CLASS:	Data type
ATTRIBUTES:	
1 Data type Name	= Integer8
2 Format	= FIXED LENGTH
2.1 Octet Length	= 1

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

5.3.1.2 Integer16

CLASS:	Data type
ATTRIBUTES:	
1 Data type Name	= Integer16
2 Format	= FIXED LENGTH
2.1 Octet Length	= 2

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

5.3.1.3 Integer24

CLASS:	Data type
ATTRIBUTES:	
1 Data type Name	= Integer24
2 Format	= FIXED LENGTH
2.1 Octet Length	= 3

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

5.3.1.4 Integer32

CLASS:	Data type
ATTRIBUTES:	
1 Data type Name	= Integer32
2 Format	= FIXED LENGTH
2.1 Octet Length	= 4

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

5.3.1.5 Unsigned8

CLASS:	Data type
ATTRIBUTES:	
1 Data type Name	= Unsigned8
2 Format	= FIXED LENGTH
2.1 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.

5.3.1.6 Unsigned16

CLASS:	Data type
ATTRIBUTES:	
1 Data type Name	= Unsigned16

2	Format	=	FIXED LENGTH
2.1	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.

5.3.1.7 Unsigned24

CLASS:	Data type
ATTRIBUTES:	
1	Data type Name = Unsigned24
2	Format = FIXED LENGTH
2.1	Octet Length = 3

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 three octets.

5.3.1.8 Unsigned32

CLASS:	Data type
ATTRIBUTES:	
1	Data type Name = Unsigned32
2	Format = FIXED LENGTH
2.1	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.

5.3.1.9 Float32

CLASS:	Data type
ATTRIBUTES:	
1	Data type Name = Float32
2	Format = FIXED LENGTH
2.1	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.

5.3.1.10 Float64

CLASS:	Data type
ATTRIBUTES:	
1	Data type Name = Float64
2	Format = FIXED LENGTH
2.1	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.

5.3.1.11 Date

CLASS:	Data type
ATTRIBUTES:	
1	Data type Name = Date
2	Format = FIXED LENGTH
2.1	Octet Length = 3

This data type consists of day, month and year minus 1900. This allows the representation of any date between 1 January 1900 and 31 December 2155.

5.3.1.12 Enumeration

CLASS:		Data type
ATTRIBUTES:		
1	Data type Name	= Enumeration
2	Format	= FIXED LENGTH
2.1	Octet Length	= 1

Data items that take on a single meaning from a list or table are encoded as Enumeration. This data type uses an unsigned integer of one octet length. The largest integer value is reserved and not used by any service.

5.3.1.13 Bit Field

CLASS:		Data type
ATTRIBUTES:		
1	Data type Name	= Bit Field
2	Format	= FIXED LENGTH
2.1	Octet Length	= 1

This data type is defined as a series of eight bits, numbered from 0 to 7. Communication of information encoded as single-bit data (such as status and diagnostic information) uses this data type.

5.3.1.14 Time

CLASS:		Data type
ATTRIBUTES:		
1	Data type Name	= Time
2	Format	= FIXED LENGTH
2.1	Octet Length	= 4

This data type is an unsigned binary integer and represents time in the increments of 1/32 of a millisecond. If this data type is used to represent time of day, then it indicates number of 1/32 of milliseconds since midnight.

5.3.1.15 Engineering unit

CLASS:		Data type
ATTRIBUTES:		
1	Data type Name	= Engineering unit
2	Format	= FIXED LENGTH
2.1	Octet Length	= 1

This type defines the measurement unit of a measured variable. The interpretation of this data type is specified by the communication profile and beyond the scope of this part of the document.

5.3.2 String Types

5.3.2.1 Packed ASCII

CLASS:		Data type
ATTRIBUTES:		
1	Data type Name	= Packed ASCII
2	Format	= STRING
2.1	Octet Length	= 1 to n

This type is a modified subset of the ASCII character code set. This subset is shown in Table 1.

Table 1 – Packed ASCII character set

Bits 4 and 5	Bits 0 to 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
2	SP ^a	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

NOTE Most significant hexadecimal digit is top to bottom; least significant is left to right.

^a SP indicates a space character.

5.3.2.2 ISO Latin-1

CLASS:

Data type

ATTRIBUTES:

- 1 Data type Name = Latin-1
 2 Format = STRING
 2.1 Octet Length = 1 to n

This type is a subset of the ISO/IEC 8859-1 (ISO Latin-1) string type. The subset is shown in Table 2.

Table 2 – ISO Latin-1 characters

Bits 4 to 7	Bits 0 to 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	SP ^a	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8																
9																
A	NBSP ^b	ı	€	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY ^c	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

NOTE Most significant hexadecimal digit is top to bottom; least significant is left to right. Grayed out cells means that no character is assigned to this code.

^a SP indicates a space character.
^b NBSP indicates a non-breaking space character.
^c SHY indicates a soft hyphen.

5.4 Data type ASE service specification

There are no operational services defined for the type object.

5.5 Summary of data types

This clause contains a summary of the defined data types as shown in Table 3.

Table 3 – Data type summary

Data type	Clause	Data type	Clause
Integer8	5.3.1.1	Float32	5.3.1.9
Integer16	5.3.1.2	Float64	5.3.1.10
Integer24	5.3.1.3	Date	5.3.1.11
Integer32	5.3.1.4	Enumeration	5.3.1.12
Unsigned8	5.3.1.5	Bit Field	5.3.1.13
Unsigned16	5.3.1.6	Time	5.3.1.15
Unsigned24	5.3.1.7	Engineering unit	5.3.1.15
Unsigned32	5.3.1.8	Packed ASCII	5.3.2.1
		ISO Latin-1	5.3.2.2

6 Communication model specification

6.1 Common parameters

Several parameters are used by more than one service. Instead of defining them with each service, the following common definitions are provided.

6.1.1 AREP ID

This parameter specifies sufficient information to identify the AREP of the remote end of the AR. One value of this parameter is reserved as broadcast address.

6.1.2 Response Code

If there is no communication error then this parameter specifies a command completion report indicating the status of the command's execution by the device. The possible values of this parameter are shown in Table 4. Its data type is Enumeration.

Table 4 – Response code values

Value	Description
Success	Command (read or Write) was executed properly.
Warning	Command (Write) was executed with the deviation as described in response (e.g., a value was set to its nearest legal value).
Error	Command (read or Write) was not executed properly. Response Code indicates the reason (e.g., the device is in Write Protect mode).

6.1.3 Application process Status

This parameter indicates the status provided by the FAL user and it is not associated with the completion of any command. Its length is one octet.

6.1.4 Extended status

This parameter indicates the VFD status. It is not associated with the completion of any command. Its length is one octet.

6.1.5 Preamble count

It specifies the minimum number of preamble octets in the DLPDU required by the destination DLE.

NOTE The value of this parameter can be obtained by using Application layer “Identify” service.

6.1.6 Communication status

This parameter specifies information about the communication failure. The possible values of this parameter are shown in Table 5.

Table 5 – Communication status values

Value	Description
Vertical parity Error	The parity of one or more of the octets received by the device was not odd.
Overrun error	At least one octet of data in the receive buffer of the PhE was overwritten before it was read (i.e., the receiver did not process incoming octet fast enough).
Framing error	The Stop bit of one or more octets received by the device was not detected by the PhE (i.e. a mark or 1 was not detected when it should have occurred).
Longitudinal parity error	The longitudinal parity calculated by the device did not match the Check octet at the end of the DLPDU.
Buffer overflow	The PhPDU or the DLPDU was too long for the receive buffer of the PhE or the DLE.
Device not available	The client did not receive any response from the server.

6.2 ASEs

6.2.1 Virtual field device ASE

6.2.1.1 Virtual field device class specification

The Virtual Field Device (VFD) is an abstract model for the description of the data and the behavior of an Application Process. VFDs contain APOs. The attributes of an APO are described by object descriptions. Services are defined for accessing a VFD's APOs, as shown in Figure 2.

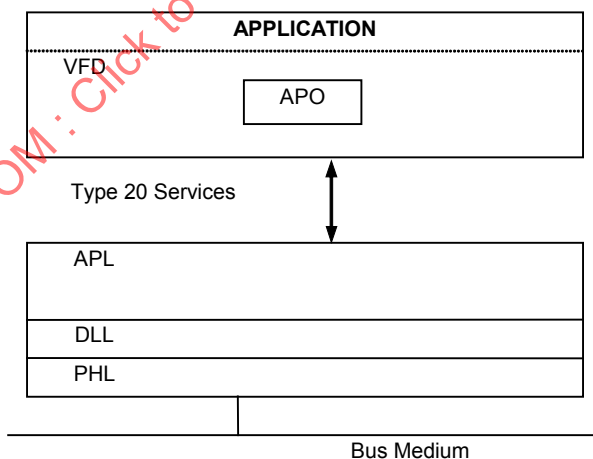


Figure 2 – VFD model

The services define no concrete interface for an implementation. They describe in an abstract form which functions may be used.

The application is not a subject of this document. It should only be indicated, how the abstractly described services may be made available to the application.

Only one VFD Object is present in a device.

6.2.1.2 Formal model

The VFD class specifies the attributes and services defined for application processes. Its parent class "top" indicates the top of the class tree.

ASE:	VFD ASE
CLASS:	VFD
CLASS ID:	—
PARENT CLASS:	TOP
1. (o) Key Attribute:	Not used
2. (m) Attribute:	Manufacturer ID
3. (m) Attribute:	Expanded Device Type
4. (m) Attribute:	Device ID
5. (m) Attribute:	Device Rev
6. (m) Attribute:	Soft Rev
7. (m) Attribute:	Hard Rev
8. (m) Attribute:	Phy Type
9. (m) Attribute:	Preamble Count
10. (m) Attribute:	Device Flag
11. (m) Attribute:	Command Rev
12. (m) Attribute:	Variable Count
13. (m) Attribute:	Config Change Counter
14. (m) Attribute:	Device ExtdStatus
15. (m) Attribute:	Distributor code
16. (m) Attribute:	Device profile
SERVICES:	
1. (m) Ops Service:	Identify

6.2.1.2.1 Attributes

Manufacturer ID

This attribute indicates the manufacturer that produced the device. A manufacturer is required to use the value assigned to it and is not permitted to use the value assigned to another manufacturer.

Expanded Device Type

This attribute indicates the manufacturer's type of the device i.e. the product name. The value of this attribute is assigned by the manufacturer. Its value specifies the set of commands and data objects supported by the device. The manufacturer is required to assigned unique value to each type of the device.

Device ID

This attribute indicates a serial number for the device. The manufacturer is required to assign a unique value for every device that has the identical values for Manufacturer ID and Device Type.

Device Rev

This attribute describes the revision level of the device. The value of this attribute is defined by the manufacturer. The value of this attribute describes the revision level of the set of commands and data objects supported by the device.

Soft Rev

This attribute indicates the revision level of the firmware in the device. The manufacturer is required to increment the value of this attribute for every new release of the device's firmware.

Hard Rev

This attribute indicates the revision level of the device hardware. The manufacturer is required to increment the value of this attribute for every major change of the device's hardware. It is not necessary to track individual hardware component changes.

Phy Type

This attribute indicates the type of Physical layer signalling used by the device.

Preamble Count

This attribute indicates the minimum number of Preambles to be sent with the request message from the Master to the Slave device.

Device Flag

This attribute indicates other information about the device such as multi-sensor, non-volatile memory control, protocol bridge, etc.

Command Rev

This attribute indicates the major revision level of the Protocol supported by the device.

Variable Count

This attribute specifies the maximum number of objects (variables) that can be accessed from the device. The value of this attribute indicates the last Variable Code that a Client application can expect to be found in the device.

Configuration Change Counter

This attribute keeps track of number of device configuration changes. The device is required to increment the value of this attribute every time it receives a request to change the configuration using application layer services, or a user of the device changes the device configuration using local means such as local operator's interface.

Device ExtdStatus

This attribute indicates the extended operational status of the device.

Distributor code

This attribute indicates the private label manufacturer that distributed the device.

Device profile

This attribute specifies the class to which the device belongs.

6.2.1.3 Identify service**6.2.1.3.1 Usage**

This service is used to request the device identification information from the AP.

6.2.1.3.2 Service primitives

The service parameters for this service are shown in Table 6.

Table 6 – Identify service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Tag	C,U	M (=)		
Result (+)			S	S (=)
Manufacturer ID			M	M (=)
Expanded Device Type			M	M (=)
Device ID			M	M (=)
Device Rev			M	M (=)
Soft Rev			M	M (=)
Hard Rev			M	M (=)
Phy Type			M	M (=)
Preamble Count			M	M (=)
Device Flag			M	M (=)
Command Rev			M	M (=)
Variable Count			M	M (=)
Configuration Change Counter			M	M (=)
Device ExtStatus			M	M (=)
Distributor code			M	M (=)
Device profile			M	M (=)
Response Code			M	M (=)
Application process Status			M	M (=)
Communication Error				S
Communication Status				M
Application process Status				C
No Match				S
Result (–)			S	S (=)
Response Code			M	M (=)
Application process Status			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter.				

Argument

The argument contains the parameters of the service request.

Tag

This parameter is used to identify the remote device. If the value of AREP ID is the Broadcast address, then this parameter is required, otherwise is is optional.

Result (+)

This selection type parameter indicates that the service request succeeded. The parameters for this selection are the attributes of VFD class object as described in 6.2.1.2.

Communication Error

This selection type parameter indicates that the service request failed due to communication error. The parameters for this selection provide the reason for the failure.

No Match

This selection is used when the indication has Tag parameter and it does not match with the Tag resident in the responding device.

Result (–)

This selection type parameter indicates that the service request failed. The parameters for this selection provide the reason for the failure.

6.2.1.3.3 Service procedure

The Confirmed Service Procedure specified in 3.5.3.3 applies to this service. If the indication primitive specifies a Tag and the responding device does not find the matching Tag then it does not send any response.

6.2.2 Variable ASE**6.2.2.1 Overview**

The Variable ASE provides services to read or write a variable object in Server device. The variable is identified by a number. This numeric identifier completely describes the data type and structure of the object. The object can be any of the types defined in 5.3.

6.2.2.2 Variable model – common**6.2.2.2.1 Simple variable****6.2.2.2.1.1 Formal model**

The simple variable object represents a single variable which is characterized by a defined Data type.

ASE:	VARIABLE ASE
CLASS:	Simple VARIABLE
PARENT CLASS:	TOP
ATTRIBUTES:	
1 (m) Key Attribute:	Numeric Identifier
2 (o) Attribute:	Variable Name
3 (m) Attribute:	Data type Name
SERVICES:	
1 (o) OpService:	Read
2 (o) OpService:	Write
3 (o) OpService:	Information report

6.2.2.2.1.2 Attributes**Numeric Identifier**

Identifies an instance of this object class.

Variable Name

It is the name assigned to the object.

Data type Name

It is the name assigned to the data type as specified in 5.3.1.

6.2.2.2.1.3 Services

Read

This service permits a client to read the value of a variable.

Write

This service permits a client to write the value of a variable.

Information report

This unconfirmed service permits a server to send value of a variable to any one or all clients.

6.2.2.2.2 Structure variable

6.2.2.2.2.1 Formal model

The structure object consists of a collection of simple variables of different data types. The record object is accessed completely and its individual fields can not be accessed separately.

ASE:	VARIABLE ASE
CLASS:	Structure VARIABLE
PARENT CLASS:	TOP
ATTRIBUTES:	
1 (m) Key Attribute:	Numeric Identifier
2 (o) Attribute:	Variable Name
3 (m) Attribute:	Number of Fields
4 (m) Attribute:	List of Fields
4.1 (o) Attribute:	Field Name
4.2 (m) Attribute:	Field Data type Name
SERVICES:	
1 (o) OpsService:	Read
2 (o) OpsService:	Write
3 (o) OpsService:	Information report

6.2.2.2.2.2 Attributes

Numeric Identifier

Identifies an instance of this object class.

Variable Name

It is the name assigned to the object.

Number of Fields

This attribute defines the number of fields in a structure.

Field Name

This optional attribute specifies the name of the field.

Field Data type Name

It is the name assigned to the data type as specified in 5.3.1.

6.2.2.2.2.3 Services

Read

This service permits a client to read the value of a variable.

Write

This service permits a client to read the value of a variable.

Information report

This unconfirmed service permits a server to send value of a variable to any one or all clients.

6.2.2.2.3 Array of simple variable

6.2.2.2.3.1 Formal model

This Array Object is used to define a constructed variable in which all elements have the identical Data type and length and are of Simple Variable class.

ASE:	VARIABLE ASE
CLASS:	Array of Simple VARIABLE
PARENT CLASS:	Simple VARIABLE
ATTRIBUTES:	
4 (m)	Attribute: Array size

6.2.2.2.3.2 Attributes

Array size

It states how many elements are contained in the Array. If the value of this attribute is "Variable" then the number of elements is not fixed.

6.2.2.3 Variable ASE service specification

6.2.2.3.1 Supported services

This subclause contains the definition of services that are unique to this ASE. The services defined for this ASE are:

- Read variable from Server device,
- Write variable to Server device, and
- Send Information report from a server device.

There are separate services for each simple or record variable, which is identified by a number. The format of the services is common to all variables. The common format of Read service is defined in 6.2.2.3.2. The common format of Write service is defined in 6.2.2.3.4. The common format of Information report service is defined in 6.2.2.3.5. The format of each variable is specified in the individual service to read or write that variable.

NOTE The service definitions in this subclause are not independent of the variable being accessed. This subclause only shows the format of the service. The service specific definitions include the variable model and the service specific parameters.

6.2.2.3.2 Primitive correlation

At the client, there is only one outstanding request for one value of AREP ID. The AREP ID is conveyed as destination address in the data link layer service primitives. Therefore, the confirm primitive is correlated with the request primitive using AREP ID. At the server, there is only one outstanding indication primitive at any time. Therefore, the response primitive is correlated with the indication primitive waiting for the response.

6.2.2.3.3 Read service – common format

6.2.2.3.3.1 Service overview

This service permits a client to read the value of a variable from the server.

6.2.2.3.3.2 Service primitives

The service parameters for this service are shown in Table 7.

Table 7 – Read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Numeric Identifier	M	M (=)	M (=)	M (=)
Additional identifier	C	M (=)	M (=)	M (=)
Subindex	C	M (=)	M (=)	M (=)
Preamble count	M			
Result (+)			S	S (=)
Variable value			M	M (=)
Response Code			M	M (=)
Application process Status			M	M (=)
Communication Error				S
Communication Status				M
Application process Status				C
Result (–)			S	S (=)
Response Code			M	M (=)
Application process Status			M	M (=)

Argument

The argument contains the parameters of the service request.

Numeric Identifier

This parameter identifies the variable. This is the Numeric Identifier assigned to the variable to be read.

Additional identifier

For some of the Numeric Identifiers, this parameter provides additional information required to identify the variable(s) to be read.

Subindex

This parameter identifies the individual element in an array variable by its position within the variable. This can be either a numeric value or an Enumeration.

Variable value

This parameter specifies the value of the variable read from the server.

Communication status

If this status is 'Device not available' then the confirm primitive does not return 'Application process status'.

6.2.2.3.4 Write service – common format

6.2.2.3.4.1 Service overview

This service permits a client to write the value of a variable to the server.

6.2.2.3.4.2 Service primitives

The service parameters for this service are shown in Table 8.

Table 8 – Write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Numeric Identifier	M	M (=)	M	M (=)
Additional identifier	C	M (=)	M	M (=)
Subindex	C	M (=)	M	M (=)
Preamble count	M			
Variable value	M	M (=)		
Result (+)			S	S (=)
Variable value			M	M (=)
Response Code			M	M (=)
Application process Status			M	M (=)
Communication Error				S
Communication Status				M
Application process Status				C
Result (–)			S	S (=)
Response Code			M	M (=)
Application process Status			M	M (=)

Argument

The argument contains the parameters of the service request.

Numeric Identifier

This parameter identifies the variable. This is the key attribute assigned to the variable to be written. It is also called command number.

Additional identifier

For some of the Numeric Identifiers, this parameter provides additional information required to identify the variable(s) to be written.

Subindex

This parameter identifies the individual element in an array variable by its position within the variable. This can be either a numeric value or an Enumeration.

Variable value

In the request primitive, this parameter specifies the desired value of the variable to be written. In the response primitive, this parameter states the actual value of the variable that was written. In some cases, the responding server writes a value different than the value in indication primitive. For example, it may round the value to the nearest allowed value or the nearest value the server can support.

NOTE The returned variable value can be used by the FAL-user to validate the write operation.

Communication status

If this status is 'Device not available' then the confirm primitive does not return 'Application process status'.

6.2.2.3.5 Information report service – common format

6.2.2.3.5.1 Service overview

This service permits a slave to send the value of a variable and master to receive it without using a request from the master. It is used to publish variable data on a cyclic basis.

6.2.2.3.5.2 Service primitives

The service parameters for this service are shown in Table 9.

Table 9 – Information report parameters

Parameter name	Req	Ind
Argument		
AREP ID	M	M (=)
Numeric Identifier	M	M (=)
Additional identifier	C	M (=)
Subindex	C	M (=)
Variable value	M	M (=)
Response Code	M	M (=)
Application process status	M	M (=)
Extended status	C	M(=)
Variable value	M	M (=)

Argument

The argument contains the parameters of the service request.

Numeric Identifier

This parameter identifies the variable. This is the key attribute assigned to the variable to be read. It is also called command number.

Additional identifier

For some of the Numeric Identifiers, this parameter provides additional information required to identify the variable(s) to be read.

Subindex

This parameter identifies the individual element in an array variable by its position within the variable. This can be either a numeric value or an Enumeration.

Variable value

This parameter specifies the value reported by the server.

6.2.3 Action ASE

6.2.3.1 Service overview

This service permits a client to command a specified action of the server. The common format of Action service is defined in this clause. For every action, there is a separate service.

NOTE The service definitions in this subclause are not independent of the action being requested. This subclause only shows the format of the service. The service specific definitions include the action behavior and the service specific parameters.

6.2.3.2 Primitive correlation

At the client, there is only one outstanding request for one value of AREP ID. The AREP ID is conveyed as destination address in the transport layer service primitives. Therefore, the confirm primitive is correlated with the request primitive using AREP ID. At the server, there is only one outstanding indication primitive at any time. Therefore, the response primitive is correlated with the indication primitive waiting for the response.

6.2.3.3 Action model

6.2.3.3.1 General

Some of the actions require a variable to validate the action. The model for such actions is defined in this subclause.

6.2.3.3.2 Formal model

ASE: Simple Action
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Key Attribute: Numeric Identifier

ASE: Qualified Action
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Key Attribute: Numeric Identifier
 2 (m) Attribute: Qualifier Name
 3 (m) Attribute: Data type Name

ASE: Simple Variable Action
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Key Attribute: Numeric Identifier
 2 (m) Attribute: Variable Name
 3 (m) Attribute: Data type Name

ASE: Structure Variable Action
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Key Attribute: Numeric Identifier
 2 (m) Attribute: Variable Name
 3 (m) Attribute: Number of Fields
 4 (m) Attribute: List of Fields
 4.1 (o) Attribute: Field Name
 4.2 (m) Attribute: Field Data type Name

6.2.3.3.3 Attributes

See 6.2.2.2 for the definition of attributes.

6.2.3.4 Action service primitives – common format

The service parameters for this service are shown in Table 10.

Table 10 – Action service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Numeric Identifier	M	M (=)	M	M (=)
Additional identifier	C	M (=)	M	M (=)
Subindex	C	M (=)	M	M (=)
Preamble count	M			
Variable value	C	M (=)		
Result (+)			S	S (=)
Variable value			S	S (=)
Response Code			M	M (=)
Application process Status			M	M (=)
Communication Error				S
Communication Status				M
Application process Status				C
Result (–)			S	S (=)
Response Code			M	M (=)
Application process Status			M	M (=)

Argument

The argument contains the parameters of the service request.

Numeric Identifier

This parameter identifies the action. It is also called command number.

Action qualifier

For some of the Numeric Identifiers, this parameter provides additional information required to identify action to be performed. If the action requires a variable and if the Numeric identifier is not sufficient, then this parameter identifies that variable.

Subindex

If the action requires an array variable, then this parameter identifies the individual element in that array variable. This parameter can be either a numeric or an enumeration.

NOTE 1 This parameter is present if the action is to add an element in the array or delete an element from the array.

Variable value

If the action requires a variable, then this parameter specifies the value of the variable. If the action is “add” then in the request parameter, this parameter specifies the desired value of the variable to be added; in the response primitive, this parameter states the actual value of the variable that was added.

NOTE 2 The returned variable value can be used by the FAL-user to validate the write operation.

Communication status

If this status is ‘Device not available’ then the confirm primitive does not return ‘Application process status’.

6.2.4 Device application services

6.2.4.1 Read primary variable

6.2.4.1.1 Service overview

This service permits a client to read the value of Primary variable and its Unit from the server.

6.2.4.1.2 Variable model

CLASS: STRUCTURE VARIABLE

ATTRIBUTES:

- | | | |
|-----|----------------------|-------------------------|
| 1 | Numeric Identifier | = 1 |
| 2 | Variable Sub Index | Not present |
| 3 | Variable Name | = Primary variable |
| 4 | Number of Fields | = 2 |
| 5 | List of Fields | |
| 5.1 | Field Name | = Primary variable Unit |
| 5.2 | Field Data type Name | = Engineering unit |
| 5.3 | Field Name | = Primary variable |
| 5.4 | Field Data type Name | = Float32 |

6.2.4.2 Read loop current and percent of range

6.2.4.2.1 Service overview

This service permits a client to read the value of the Loop current and its associated Percent of range from the server.

6.2.4.2.2 Variable model

CLASS: STRUCTURE VARIABLE

ATTRIBUTES:

- | | | |
|-----|----------------------|--------------------|
| 1 | Numeric Identifier | = 2 |
| 2 | Variable Sub Index | Not present |
| 3 | Variable Name | = Loop current |
| 4 | Number of Fields | = 2 |
| 5 | List of Fields | |
| 5.1 | Field Name | = Loop current |
| 5.2 | Field Data type Name | = Float32 |
| 5.3 | Field Name | = Percent of range |
| 5.4 | Field Data type Name | = Float32 |

6.2.4.3 Read dynamic variables and loop current

6.2.4.3.1 Service overview

This service permits a client to read the value of the Loop current and up to four predefined Dynamic variables from the server. The number of variables in the response data depends upon the type of the device.

6.2.4.3.2 Variable model

CLASS: STRUCTURE VARIABLE

ATTRIBUTES:

- | | | |
|---|--------------------|-------------|
| 1 | Numeric Identifier | = 3 |
| 2 | Variable Sub Index | Not present |

- 3 Variable Name = Dynamic variables
- 4 List of Fields
 - 4.1 (c) Constraint Number of fields = {3 | 5 | 7 | 9}
 - 4.1.1 (m) Field Name = Loop current
 - 4.1.2 (m) Field Data type Name = Float32
 - 4.1.3 (m) Field Name = Primary variable Unit
 - 4.1.4 (m) Field Data type Name = Engineering unit
 - 4.1.5 (m) Field Name = Primary variable
 - 4.1.6 (m) Field Data type Name = Float32
 - 4.2 (c) Constraint Number of Fields = {5 | 7 | 9}
 - 4.2.1 (m) Field Name = Secondary variable Unit
 - 4.2.2 (m) Field Data type Name = Engineering unit
 - 4.2.3 (m) Field Name = Secondary variable
 - 4.2.4 (m) Field Data type Name = Float32
 - 4.3 (c) Constraint Number of Fields = {7 | 9}
 - 4.3.1 (m) Field Name = Tertiary variable Unit
 - 4.3.2 (m) Field Data type Name = Engineering unit
 - 4.3.3 (m) Field Name = Tertiary variable
 - 4.3.4 (m) Field Data type Name = Float32
 - 4.4 (c) Constraint Number of Fields = 9
 - 4.4.1 (m) Field Name = Quaternary variable Unit
 - 4.4.2 (m) Field Data type Name = Engineering unit
 - 4.4.3 (m) Field Name = Quaternary variable
 - 4.4.4 (m) Field Data type Name = Float32

The number of fields can be either 3 or 5 or 7 or 9. The fields marked as 4.1.1 to 4.1.6 are always present. If the number of fields is 5 or 7 or 9, then fields 4.2.1 to 4.2.4 are present. If the number of fields is 7 or 9, then fields 4.3.1 to 4.3.4 are present. If the number of fields is 9, then fields 4.4.1 to 4.4.4 are present.

6.2.4.4 Write loop configuration

6.2.4.4.1 Service overview

This service permits a client to write the value of the Polling address and the Loop current mode to the server.

6.2.4.4.2 Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

- 1 Numeric Identifier = 6
- 2 Variable Sub Index Not present
- 3 Variable Name = Loop configuration
- 4 Number of Fields = 2
- 5 List of Fields
 - 5.1 Field Name = Polling address
 - 5.2 Field Data type Name = Unsigned8
 - 5.3 Field Name = Loop current mode
 - 5.4 Field Data type Name = Enumeration

6.2.4.5 Read loop configuration

6.2.4.5.1 Service overview

This service permits a client to read the value of the Polling address and the Loop current mode from the server.

6.2.4.5.2 Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 7
2	Variable Sub Index	Not present
3	Variable Name	= Loop configuration
4	Number of Fields	= 2
5	List of Fields	
5.1	Field Name	= Polling address
5.2	Field Data type Name	= Unsigned8
5.3	Field Name	= Loop current mode
5.4	Field Data type Name	= Enumeration

6.2.4.6 Read dynamic variable families classifications

6.2.4.6.1 Service overview

This service permits a client to read the value of the Classification associated with the Dynamic variables.

6.2.4.6.2 Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 8
2	Variable Sub Index	Not present
3	Variable Name	= Dynamic variables classification
4	Number of Fields	= 4
5	List of Fields	
5.1	Field Name	= Primary variable classification
5.2	Field Data type Name	= Enumeration
5.3	Field Name	= Secondary variable classification
5.4	Field Data type Name	= Enumeration
5.5	Field Name	= Tertiary variable classification
5.6	Field Data type Name	= Enumeration
5.7	Field Name	= Quaternary variable classification
5.8	Field Data type Name	= Enumeration

6.2.4.7 Read device variables with status

6.2.4.7.1 Service overview

This service permits a client to read the value of up to eight Device or Dynamic variable from the server. Client may request any number of Device variables. The variables are identified by assigning the Variable code to Variable sub index. The number of variables in the response data matches the number of variables in the request.

6.2.4.7.2 Request Variable Sub Index model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

- 1 Numeric Identifier = 9
- 2 Variable Name = Device Variables
- 3 (c) Constraint Variable Sub Index = up to 8
- 3.1 (m) Sub Index 1
- 3.1.1 (m) Field Name = Variable code
- 3.1.2 (m) Field Data type Name = Unsigned8
- 3.2 (c) Sub Index 2
- 3.2.1 (m) Field Name = Variable code
- 3.2.2 (m) Field Data type Name = Unsigned8
- 3.3 (c) Sub Index 3
- 3.3.1 (m) Field Name = Variable code
- 3.3.2 (m) Field Data type Name = Unsigned8
- 3.4 (c) Sub Index 4
- 3.4.1 (m) Field Name = Variable code
- 3.4.2 (m) Field Data type Name = Unsigned8
- 3.5 (c) Sub Index 5
- 3.5.1 (m) Field Name = Variable code
- 3.5.2 (m) Field Data type Name = Unsigned8
- 3.6 (c) Sub Index 6
- 3.6.1 (m) Field Name = Variable code
- 3.6.2 (m) Field Data type Name = Unsigned8
- 3.7 (c) Sub Index 7
- 3.7.1 (m) Field Name = Variable code
- 3.7.2 (m) Field Data type Name = Unsigned8
- 3.8 (c) Sub Index 8
- 3.8.1 (m) Field Name = Variable code
- 3.8.2 (m) Field Data type Name = Unsigned8

6.2.4.7.3 Response Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

- 1 Numeric Identifier = 9
- 2 Variable Name = Device Variables
- 3 List of Fields
- 3.1 (c) Constraint Number of Variables = {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8}
- 3.1.1 (m) Field Name = Device ExtStatus
- 3.1.2 (m) Field Data type Name = Bit Field
- 3.1.3 (m) Field Name = Variable code
- 3.1.4 (m) Field Data type Name = Unsigned8
- 3.1.5 (m) Field Name = Variable class
- 3.1.6 (m) Field Data type Name = Enumeration
- 3.1.7 (m) Field Name = Variable Unit
- 3.1.8 (m) Field Data type Name = Engineering unit
- 3.1.9 (m) Field Name = Variable
- 3.1.10 (m) Field Data type Name = Float32
- 3.1.11 (m) Field Name = Variable status
- 3.1.12 (m) Field Data type Name = Bit Field
- 3.2 (c) Constraint Number of Variables = {2 | 3 | 4 | 5 | 6 | 7 | 8}
- 3.2.1 (m) Field Name = Variable code

- 3.2.2 (m) Field Data type Name = Unsigned8
- 3.2.3 (m) Field Name = Variable class
- 3.2.4 (m) Field Data type Name = Enumeration
- 3.2.5 (m) Field Name = Variable Unit
- 3.2.6 (m) Field Data type Name = Engineering unit
- 3.2.7 (m) Field Name = Variable
- 3.2.8 (m) Field Data type Name = Float32
- 3.2.9 (m) Field Name = Variable status
- 3.2.10 (m) Field Data type Name = Bit Field
- 3.3 (c) Constraint Number of Variables = {3 | 4 | 5 | 6 | 7 | 8}
- 3.3.1 (m) Field Name = Variable code
- 3.3.2 (m) Field Data type Name = Unsigned8
- 3.3.3 (m) Field Name = Variable class
- 3.3.4 (m) Field Data type Name = Enumeration
- 3.3.5 (m) Field Name = Variable Unit
- 3.3.6 (m) Field Data type Name = Engineering unit
- 3.3.7 (m) Field Name = Variable
- 3.3.8 (m) Field Data type Name = Float32
- 3.3.9 (m) Field Name = Variable status
- 3.3.10 (m) Field Data type Name = Bit Field
- 3.4 (c) Constraint Number of Variables = {4 | 5 | 6 | 7 | 8}
- 3.4.1 (m) Field Name = Variable code
- 3.4.2 (m) Field Data type Name = Unsigned8
- 3.4.3 (m) Field Name = Variable class
- 3.4.4 (m) Field Data type Name = Enumeration
- 3.4.5 (m) Field Name = Variable Unit
- 3.4.6 (m) Field Data type Name = Engineering unit
- 3.4.7 (m) Field Name = Variable
- 3.4.8 (m) Field Data type Name = Float32
- 3.4.9 (m) Field Name = Variable status
- 3.4.10 (m) Field Data type Name = Bit Field
- 3.5 (c) Constraint Number of Variables = {5 | 6 | 7 | 8}
- 3.5.1 (m) Field Name = Variable code
- 3.5.2 (m) Field Data type Name = Unsigned8
- 3.5.3 (m) Field Name = Variable class
- 3.5.4 (m) Field Data type Name = Enumeration
- 3.5.5 (m) Field Name = Variable Unit
- 3.5.6 (m) Field Data type Name = Engineering unit
- 3.5.7 (m) Field Name = Variable
- 3.5.8 (m) Field Data type Name = Float32
- 3.5.9 (m) Field Name = Variable status
- 3.5.10 (m) Field Data type Name = Bit Field
- 3.6 (c) Constraint Number of Variables = {6 | 7 | 8}
- 3.6.1 (m) Field Name = Variable code
- 3.6.2 (m) Field Data type Name = Unsigned8
- 3.6.3 (m) Field Name = Variable class
- 3.6.4 (m) Field Data type Name = Enumeration
- 3.6.5 (m) Field Name = Variable Unit
- 3.6.6 (m) Field Data type Name = Engineering unit
- 3.6.7 (m) Field Name = Variable
- 3.6.8 (m) Field Data type Name = Float32
- 3.6.9 (m) Field Name = Variable status
- 3.6.10 (m) Field Data type Name = Bit Field

3.7	(c)	Constraint	Number of Variables = {7 8}
3.7.1	(m)	Field Name	= Variable code
3.7.2	(m)	Field Data type Name	= Unsigned8
3.7.3	(m)	Field Name	= Variable class
3.7.4	(m)	Field Data type Name	= Enumeration
3.7.5	(m)	Field Name	= Variable Unit
3.7.6	(m)	Field Data type Name	= Engineering unit
3.7.7	(m)	Field Name	= Variable
3.7.8	(m)	Field Data type Name	= Float32
3.7.9	(m)	Field Name	= Variable status
3.7.10	(m)	Field Data type Name	= Bit Field
3.8	(c)	Constraint	Number of Variables = 8
3.8.1	(m)	Field Name	= Variable code
3.8.2	(m)	Field Data type Name	= Unsigned8
3.8.3	(m)	Field Name	= Variable class
3.8.4	(m)	Field Data type Name	= Enumeration
3.8.5	(m)	Field Name	= Variable Unit
3.8.6	(m)	Field Data type Name	= Engineering unit
3.8.7	(m)	Field Name	= Variable
3.8.8	(m)	Field Data type Name	= Float32
3.8.9	(m)	Field Name	= Variable status
3.8.10	(m)	Field Data type Name	= Bit Field
3.9	(m)	Field Name	= Slot 0 time stamp
3.10	(m)	Field Data type Name	= Time

The number of variables can be any number from 1 to 8. The fields marked as 3.1.1 to 3.1.12 and 3.9 to 3.10 are always present.

6.2.4.8 Read message

6.2.4.8.1 Service overview

This service permits a client to read the value of the Message string from the server.

6.2.4.8.2 Variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

1	Numeric Identifier	= 12
2	Variable Sub Index	Not present
3	Variable Name	= Message
4	Field Name	= Message
5	Field Data type Name	= Packed ASCII

6.2.4.9 Read tag, descriptor and date

6.2.4.9.1 Service overview

This service permits a client to read the value of the Tag, Descriptor and Date from the server.

6.2.4.9.2 Variable model

CLASS: STRUCTURE VARIABLE

ATTRIBUTES:

1	Numeric Identifier	= 13
---	--------------------	------

2	Variable Sub Index	Not present
3	Variable Name	= Tag Descriptor
4	Number of Fields	= 3
5	List of Fields	
5.1	Field Name	= Tag
5.2	Field Data type Name	= Packed ASCII
5.3	Field Name	= Descriptor
5.4	Field Data type Name	= Packed ASCII
5.5	Field Name	= Date
5.6	Field Data type Name	= Date

6.2.4.10 Read primary variable transducer information

6.2.4.10.1 Service overview

This service permits a client to read the value of the Transducer serial number, Limits/Minimum span Unit, Upper transducer limit, Lower transducer limit, and Minimum span for the Primary variable transducer from the server.

6.2.4.10.2 Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 14
2	Variable Sub Index	Not present
3	Variable Name	= Primary variable transducer Information
4	Number of Fields	= 5
5	List of Fields	
5.1	Field Name	= Transducer serial number
5.2	Field Data type Name	= Unsigned24
5.3	Field Name	= Transducer limits and minimum span Units
5.4	Field Data type Name	= Engineering unit
5.5	Field Name	= Upper transducer limit
5.6	Field Data type Name	= Float32
5.7	Field Name	= Lower transducer limit
5.8	Field Data type Name	= Float32
5.9	Field Name	= Minimum span
5.10	Field Data type Name	= Float32

6.2.4.11 Read device information

6.2.4.11.1 Service overview

This service permits a client to read the value of the PV alarm selection code, PV transfer function, PV upper and lower range values Unit, PV upper range, PV lower range, PV damping, Write protect, Private label distributor and PV analog channel from the server.

6.2.4.11.2 Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 15
2	Variable Sub Index	Not present
3	Variable Name	= Device information
4	Number of Fields	= 9
5	List of Fields	

- | | | |
|------|----------------------|------------------------|
| 5.1 | Field Name | = PV alarm selection |
| 5.2 | Field Data type Name | = Enumeration |
| 5.3 | Field Name | = PV transfer function |
| 5.4 | Field Data type Name | = Enumeration |
| 5.5 | Field Name | = PV Range Unit |
| 5.6 | Field Data type Name | = Engineering unit |
| 5.7 | Field Name | = PV upper range |
| 5.8 | Field Data type Name | = Float32 |
| 5.9 | Field Name | = PV lower range |
| 5.10 | Field Data type Name | = Float32 |
| 5.11 | Field Name | = PV damping |
| 5.12 | Field Data type Name | = Float32 |
| 5.13 | Field Name | = Write protect |
| 5.14 | Field Data type Name | = Enumeration |
| 5.15 | Field Name | = Reserved |
| 5.16 | Field Data type Name | = Enumeration |
| 5.17 | Field Name | = PV analog channel |
| 5.18 | Field Data type Name | = Bit Field |

6.2.4.12 Read final assembly number

6.2.4.12.1 Service overview

This service permits a client to read the value of the Final assembly number associated with the device from the server.

6.2.4.12.2 Variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

- | | | |
|---|----------------------|-------------------------|
| 1 | Numeric Identifier | = 16 |
| 2 | Variable Sub Index | Not present |
| 3 | Variable Name | = Final assembly number |
| 4 | Field Name | = Final assembly number |
| 5 | Field Data type Name | = Unsigned24 |

6.2.4.13 Write message

6.2.4.13.1 Service overview

This service permits a client to write the value of the Message string to the server.

6.2.4.13.2 Variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

- | | | |
|---|----------------------|----------------|
| 1 | Numeric Identifier | = 17 |
| 2 | Variable Sub Index | Not present |
| 3 | Variable Name | = Message |
| 4 | Field Name | = Message |
| 5 | Field Data type Name | = Packed ASCII |

6.2.4.14 Write tag, descriptor and date**6.2.4.14.1 Service overview**

This service permits a client to write the value of the Tag, Descriptor and Date to the server.

6.2.4.14.2 Variable model

CLASS: **STRUCTURE VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 18
2	Variable Sub Index	Not present
3	Variable Name	= Tag Descriptor
4	Number of Fields	= 3
5	List of Fields	
5.1	Field Name	= Tag
5.2	Field Data type Name	= Packed ASCII
5.3	Field Name	= Descriptor
5.4	Field Data type Name	= Packed ASCII
5.5	Field Name	= Date
5.6	Field Data type Name	= Date

6.2.4.15 Write final assembly number**6.2.4.15.1 Service overview**

This service permits a client to write the value of the Final assembly number associated with the device to the server.

6.2.4.15.2 Variable model

CLASS: **Simple VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 19
2	Variable Sub Index	Not present
3	Variable Name	= Final assembly number
4	Field Name	= Final assembly number
5	Field Data type Name	= Unsigned24

6.2.4.16 Read long tag**6.2.4.16.1 Service overview**

This service permits a client to read the value of the Long tag from the server.

6.2.4.16.2 Variable model

CLASS: **Simple VARIABLE**

ATTRIBUTES:

1	Numeric Identifier	= 20
2	Variable Sub Index	Not present
3	Variable Name	= Long tag
4	Field Name	= Long tag
5	Field Data type Name	= Latin-1

6.2.4.17 Write long tag

6.2.4.17.1 Service overview

This service permits a client to write the value of the Long tag from the server.

6.2.4.17.2 Variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

1	Numeric Identifier	= 22
2	Variable Sub Index	Not present
3	Variable Name	= Long tag
4	Field Name	= Long tag
5	Field Data type Name	= Latin-1

6.2.4.18 Reset configuration changed flag

This service is specified in IEC 62591:2010, 7.3.2.4.18.

6.2.4.19 Perform self test

This service is specified in IEC 62591:2010, 7.3.2.4.19.

6.2.4.20 Perform device reset

This service is specified in IEC 62591:2010, 7.3.2.4.20.

6.2.4.21 Read additional device status

This service is specified in IEC 62591:2010, 7.3.2.4.21.

6.2.4.22 Reset more status available

This service is specified in IEC 62591:2010, 7.3.2.4.22.

6.2.4.23 Read device variable information

This service is specified in IEC 62591:2010, 7.3.2.4.23.

6.2.4.24 Write device variable

This service is specified in IEC 62591:2010, 7.3.2.4.24.

6.2.4.25 Write primary variable range values

6.2.4.25.1 Service overview

This service sets the relationship between the loop current 4,00 mA, 20,0 mA points and the primary variable value. The upper range value of the primary variable is independent of the lower range value and the upper range value can be smaller than the lower range value.

6.2.4.25.2 Variable model

CLASS: STRUCTURE VARIABLE

ATTRIBUTES:

1	Numeric Identifier	= 35
2	Variable Name	= Primary variable range

- 3 Number of Fields = 3
- 4 List of Fields
- 4.1 Field Name = Range unit
- 4.2 Field Data type Name = Engineering unit
- 4.3 Field Name = Upper range
- 4.4 Field Data type Name = Float32
- 4.5 Field Name = Lower range
- 4.6 Field Data type Name = Float32

6.2.4.26 Enter-exit fixed current mode

6.2.4.26.1 Service overview

This service is used to put the device in fixed Loop current mode or exit from the fixed Loop current mode.

6.2.4.26.2 Request variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

- 1 Numeric Identifier = 40
- 2 Variable Name = Fixed current level
- 3 Variable Data type Name = Float32

6.2.4.26.3 Response variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

- 4 Numeric Identifier = 40
- 5 Variable Name = Actual current level
- 6 Variable Data type Name = Float32

6.2.4.27 Write primary variable unit

6.2.4.27.1 Service overview

This service is used to set the measurement unit for the Primary variable.

6.2.4.27.2 Variable model

CLASS: Simple VARIABLE

ATTRIBUTES:

- 1 Numeric Identifier = 44
- 2 Variable Name = Primary variable unit code
- 3 Variable Data type Name = Engineering unit

6.2.4.28 Trim loop current zero

6.2.4.28.1 Service overview

This service is used to trim the internal zero calibration, so that the Loop current matches the value measured by the Master.

6.2.4.28.2 Variable model**CLASS:** Simple VARIABLE**ATTRIBUTES:**

- 1 Numeric Identifier = 45
- 2 Variable Name = Loop current level
- 3 Variable Data type Name = Float32

6.2.4.29 Trim loop current gain**6.2.4.29.1 Service overview**

This service is used to trim the internal gain calibration, so that the Loop current matches the value measured by the Master.

6.2.4.29.2 Variable model**CLASS:** Simple VARIABLE**ATTRIBUTES:**

- 1 Numeric Identifier = 46
- 2 Variable Name = Loop current level
- 3 Variable Data type Name = Float32

6.2.4.30 Read dynamic variable assignment**6.2.4.30.1 Service overview**

This service reads the Device variable numbers that are assigned to the Primary, Secondary, Tertiary and Quaternary variables.

6.2.4.30.2 Variable model**CLASS:** STRUCTURE VARIABLE**ATTRIBUTES:**

- 1 Numeric Identifier = 50
- 2 Variable Name = Dynamic variable assignment
- 3 Number of Fields = 4
- 4 List of Fields
 - 4.1 Field Name = Primary variable assignment
 - 4.2 Field Data type Name = Unsigned8
 - 4.3 Field Name = Secondary variable assignment
 - 4.4 Field Data type Name = Unsigned8
 - 4.5 Field Name = Tertiary variable assignment
 - 4.6 Field Data type Name = Unsigned8
 - 4.7 Field Name = Quaternary variable assignment
 - 4.8 Field Data type Name = Unsigned8

6.2.4.31 Write dynamic variable assignment**6.2.4.31.1 Service overview**

This service writes the Device variable numbers that are assigned to the Primary, Secondary, Tertiary and Quaternary variables.

6.2.4.31.2 Variable model

CLASS:	STRUCTURE VARIABLE
ATTRIBUTES:	
1 Numeric Identifier	= 51
2 Variable Name	= Dynamic variable assignment
3 Number of Fields	= 4
4 List of Fields	
4.1 Field Name	= Primary variable assignment
4.2 Field Data type Name	= Unsigned8
4.3 Field Name	= Secondary variable assignment
4.4 Field Data type Name	= Unsigned8
4.5 Field Name	= Tertiary variable assignment
4.6 Field Data type Name	= Unsigned8
4.7 Field Name	= Quaternary variable assignment
4.8 Field Data type Name	= Unsigned8

6.2.4.32 Write number of response preambles**6.2.4.32.1 Service overview**

This service is used to set the number of preamble characters to be sent by a Slave device before the start of a response PhPDU.

6.2.4.32.2 Variable model

CLASS:	Simple VARIABLE
ATTRIBUTES:	
1 Numeric Identifier	= 59
2 Variable Name	= Preamble count
3 Variable Data type Name	= Unsigned8

6.2.4.33 Read device variable trim points**6.2.4.33.1 Service overview**

This service reads the last successful trim points of the Device variable.

6.2.4.33.2 Request and response parameters

CLASS:	STRUCTURE VARIABLE
ATTRIBUTES:	
1 Numeric Identifier	= 80
2 Subindex	
2.1 Subindex Name	= Variable code
2.2 Subindex Data type	= Unsigned8

6.2.4.33.3 Variable model

CLASS:	Array of Structure VARIABLE
ATTRIBUTES:	
1 Numeric Identifier	= 80
2 Variable Name	= Device variable trim points record
3 Number of Fields	= 3
4 List of Fields	
4.1 Field Name	= Trim point unit

- 4.2 Field Data type Name = Engineering unit
- 4.3 Field Name = Lower trim point
- 4.4 Field Data type Name = Float32
- 4.5 Field Name = Upper trim point
- 4.6 Field Data type Name = Float32
- 5 Array size = Variable

6.2.4.34 Read device variable trim guidelines

6.2.4.34.1 Service overview

This service reads the information that a Host will need to guide a user through a correct selection of trim points.

6.2.4.34.2 Request and response parameters

CLASS: STRUCTURE VARIABLE

ATTRIBUTES:

- 1 Numeric Identifier = 81
- 2 Subindex
- 2.1 Subindex Name = Variable code
- 2.2 Subindex Data type = Unsigned8

6.2.4.34.3 Variable model

CLASS: Array of Structure VARIABLE

ATTRIBUTES:

- 1 Numeric Identifier = 81
- 2 Variable Name = Device variable trim guidelines record
- 3 Number of Fields = 7
- 4 List of Fields
 - 4.1 Field Name = Trim points supported
 - 4.2 Field Data type Name = Enumeration
 - 4.3 Field Name = Trim point unit
 - 4.4 Field Data type Name = Engineering unit
 - 4.5 Field Name = Minimum lower trim point
 - 4.6 Field Data type Name = Float32
 - 4.7 Field Name = Maximum lower trim point
 - 4.8 Field Data type Name = Float32
 - 4.9 Field Name = Minimum upper trim point
 - 4.10 Field Data type Name = Float32
 - 4.11 Field Name = Maximum upper trim point
 - 4.12 Field Data type Name = Float32
 - 4.13 Field Name = Minimum trim point difference
 - 4.14 Field Data type Name = Float32
- 5 Array size = Variable

6.2.4.35 Write device variable trim point

6.2.4.35.1 Service overview

This service writes the value of one trim point for a Device variable to perform a calibration adjustment.

6.2.4.35.2 Request and response parameters**CLASS:** STRUCTURE VARIABLE**ATTRIBUTES:**

- | | | |
|-----|--------------------|-----------------|
| 1 | Numeric Identifier | = 82 |
| 2 | Subindex | |
| 2.1 | Subindex Name | = Variable code |
| 2.2 | Subindex Data type | = Unsigned8 |

6.2.4.35.3 Variable model**CLASS:** Array of Structure VARIABLE**ATTRIBUTES:**

- | | | |
|-----|----------------------|-------------------------------------|
| 1 | Numeric Identifier | = 82 |
| 2 | Variable Name | = Device variable trim point record |
| 3 | Number of Fields | = 3 |
| 4 | List of Fields | |
| 4.1 | Field Name | = Trim points type |
| 4.2 | Field Data type Name | = Enumeration |
| 4.3 | Field Name | = Trim point unit |
| 4.4 | Field Data type Name | = Engineering unit |
| 4.5 | Field Name | = Trim point value |
| 4.6 | Field Data type Name | = Float32 |
| 5 | Array size | = Variable |

6.2.4.36 Reset device variable trim**6.2.4.36.1 Service overview**

This service resets the value of trim points for a Device variable to the default factory values.

6.2.4.36.2 Request and response parameters**CLASS:** Simple VARIABLE**ATTRIBUTES:**

- | | | |
|---|-------------------------|-----------------|
| 1 | Numeric Identifier | = 83 |
| 2 | Variable Name | = Variable code |
| 3 | Variable Data type Name | = Unsigned8 |

6.2.4.37 Write publish data period

This service is specified in IEC 62591:2010, 7.3.2.4.26.

6.2.4.38 Write publish data trigger

This service is specified in IEC 62591:2010, 7.3.2.4.27.

6.2.4.39 Read publish data mode configuration

This service is specified in IEC 62591:2010, 7.3.2.4.28.

6.2.4.40 Flush delayed responses

This service is specified in IEC 62591:2010, 7.3.2.4.29.

6.2.4.41 Write publish data device variables

This service is specified in IEC 62591:2010, 7.3.2.4.30.

6.2.4.42 Write publish data mode command number

This service is specified in IEC 62591:2010, 7.3.2.4.31.

6.2.4.43 Write publish data mode control

This service is specified in IEC 62591:2010, 7.3.2.4.32.

6.3 ARs

6.3.1 Application relationship ASE

6.3.1.1 Overview

ARs are composed of a set of endpoints of compatible classes. One endpoint of each AR has to be Master class and the other end has to be Slave class. Each device can have only one instance of an Slave class AREP.

A device can use more than one Master class AREP to communicate with several Slave class AREPs. But, only one AR is active at any time. The user at Master class AREP provides the identification of the Slave class AREP as a parameter in the request primitive.

6.3.1.2 Master class AREP

6.3.1.2.1 Formal model

ASE: **AR ASE**
CLASS: **Master**
CLASS ID: **Not used**
PARENT CLASS: **Top**
ATTRIBUTES:
 1 (m) Attribute: Address
SERVICES:
 1 (m) OpService: AR-Get Attribute
 2 (m) OpService: AR-Set Attribute

6.3.1.2.2 Attributes

Address

This attribute specifies the Data Link Address of Master device.

6.3.1.3 Slave class AREP

6.3.1.3.1 Formal model

ASE: **AR ASE**
CLASS: **Slave**
CLASS ID: **Not used**
PARENT CLASS: **Top**
ATTRIBUTES:
 1 (m) Attribute: PollAddress
 2 (m) Attribute: UniqueAddress
 3 (m) Attribute: PreambleLength
SERVICES:
 1 (m) OpService: AR-Get Attribute

2 (m) OpsService: AR-Set Attribute

6.3.1.3.2 Attributes

PollAddress

This attribute specifies the Data Link Address of Slave device. The device responds only to this address in an Identify frame received from a Master device.

Unique address

This attribute specifies the Data Link Address of Slave device. The device responds only to this address in all frames received from a Master device. An all zero value of this attribute is reserved as Broadcast address.

PreambleLength

This attribute specifies the number of preamble octets that have to be transmitted in the beginning of every request PhPDU from Master to this Slave.

6.3.1.4 Application relationship ASE service specifications

6.3.1.4.1 Overview

This subclause contains the definition of the services that are unique to this ASE. The services are

AR Get Attribute

AR Set Attribute

6.3.1.4.2 AR-Get attribute service

6.3.1.4.2.1 Overview

This confirmed service is used to read the value of attributes of an AREP locally.

6.3.1.4.2.2 Service parameters

The service parameters for the AR Get Attribute Service are shown in Table 11.

Table 11 – AR get attributes service parameters

Parameter name	Req	Cnf
Argument	M	
Attribute index	M	
Result (+)		S
Value		C
Result (-)		S
Status		C
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

Argument

The argument carries the parameters of the service request.

Attribute index

This parameter identifies the attribute.

Result (+)

This selection type parameter indicates that the request succeeded.

Value

This parameter is returned with the requested attribute value if the request succeeded.

Result (-)

This selection type parameter indicates that the request failed.

Status

If the request failed, Status is returned indicating the reason for failure. The possible values are: Illegal attribute index, Unknown value.

6.3.1.4.3 AR Set attributes service

This confirmed service is used to set the current value of attributes of an AREP locally.

6.3.1.4.3.1 Service parameters

The service parameters for the AR Set Attribute Service are shown in Table 12.

Table 12 – AR set attributes service parameters

Parameter name	Req	Cnf
Argument	M	
Attribute index	M	
Value	M	
Result		
Status		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter.		

Argument

The argument carries the parameters of the service request.

Attribute index

This parameter identifies the attribute to be updated.

Value

This parameter holds the new attribute value.

Status

As a result of the request, Status is returned indicating OK or the reason for failure. The possible values are: OK, Illegal attribute index, Illegal value.

6.4 Summary of classes

This subclause contains a summary of the defined Classes. Table 13 provides a summary of the classes.

Table 13 – Class summary

ASE	Class
Virtual field device	VFD
Data type	Fixed Length Data type
	String Data type
	Structure
	Array
Variable	Simple Variable
	Structure Variable
	Array

6.5 Permitted services by AREP role

Table 14 and Table 15 define the valid combinations of services and AREP class (which service APDUs and AREP with the specified class can send or receive).

Table 14 – Confirmed services by AREP class

Services	Master		Slave	
	req	cnf	ind	rsp
VFD ASE				
Identify	X	X	X	X
Variable ASE				
Read service – all	X	X	X	X
Write service – all	X	X	X	X
Action ASE	X	X	X	X

Table 15 – Unconfirmed services by AREP class

Services	Slave	Master
	req	ind
Variable ASE		
Information report	X	X

Bibliography

IEC 61158-6-20, *Industrial communication networks – Fieldbus specifications – Part 6-20: Application layer protocol specification – Type 20 elements*

IEC 61784-1, *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

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

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

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014

SOMMAIRE

AVANT-PROPOS	59
INTRODUCTION.....	61
1 Domaine d'application	62
2 Références normatives	62
3 Termes, définitions, symboles, abréviations et conventions	63
3.1 Termes et définitions d'autres normes ISO/CEI	63
3.2 Termes de la CEI 61158-1	64
3.3 Définitions spécifiques à la couche application de bus de terrain de type 20	66
3.4 Abréviations et symboles.....	68
3.5 Conventions	69
4 Concepts	72
5 ASE des types de données.....	72
5.1 Vue d'ensemble.....	72
5.2 Définition formelle des objets de types de données	75
5.3 Types de données définis par la FAL.....	76
5.4 Spécification des services ASE pour le type de données	80
5.5 Synthèse des types de données	80
6 Spécification du modèle de Communication.....	80
6.1 Paramètres communs.....	80
6.2 ASE.....	82
6.3 AR.....	109
6.4 Synthèse des classes.....	112
6.5 Services admis par le rôle de l'AREP	112
Bibliographie.....	114
Figure 1 – Hiérarchie des classes de types de données.....	73
Figure 2 – Modèle VFD	82
Tableau 1 – Ensemble de caractères ASCII compact	79
Tableau 2 – Caractères ISO Latin n°1	80
Tableau 3 – Synthèse des types de données	80
Tableau 4 – Valeurs de code de réponse	81
Tableau 5 – Valeurs d'état de communication	81
Tableau 6 – Paramètres de service d'identification	85
Tableau 7 – Paramètres de service Lecture	89
Tableau 8 – Paramètres de service Ecriture.....	90
Tableau 9 – Paramètres de service Rapport d'information	91
Tableau 10 – Paramètres de service Action	93
Tableau 11 – Paramètres de service AR get attributes.....	111
Tableau 12 – Paramètres de service AR set attributes	111
Tableau 13 – Synthèse des classes	112
Tableau 14 – Services confirmés par classe d'AREP	112
Tableau 15 – Services non confirmés par classe d'AREP	113

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

**RÉSEAUX DE COMMUNICATION INDUSTRIELS –
SPÉCIFICATIONS DES BUS DE TERRAIN –****Partie 5-20: Définition des services de la couche application –
Éléments de type 20**

AVANT-PROPOS

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

L'attention est attirée sur le fait que l'utilisation du type de protocole associé est restreinte par les détenteurs des droits de propriété intellectuelle. En tout état de cause, l'engagement de renonciation partielle aux droits de propriété intellectuelle pris par les détenteurs de ces droits autorise l'utilisation d'un type de protocole de couche avec les autres protocoles de couche du même type, ou dans des combinaisons avec d'autres types autorisées explicitement par les détenteurs des droits de propriété intellectuelle pour ce type.

NOTE Les combinaisons de types de protocoles sont spécifiées dans la CEI 61784-1 et la CEI 61784-2.

La Norme internationale CEI 61158-5-20 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de la CEI: Mesure, commande et automation dans les processus industriels.

Cette troisième édition annule et remplace la deuxième édition parue en 2010. Cette édition constitue une révision technique.

Les modifications majeures par rapport à l'édition précédente sont mentionnées ci-dessous:

- ajout de Types de données;
- ajout de services;
- mise à jour des Références Normatives, des Termes, définitions, symboles, abréviations.
- correction des erreurs rédactionnelles et syntaxiques.

Le texte de cette norme est issu des documents suivants:

FDIS	Rapport de vote
65C/763/FDIS	65C/773/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

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

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

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

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

INTRODUCTION

Le présent document fait partie d'une série élaborée pour faciliter l'interconnexion des composants de systèmes d'automatisation. Il est apparenté à d'autres documents dans l'ensemble, comme défini par le modèle de référence de bus de terrain "à trois couches" décrit dans la CEI 61158-1.

Le service d'application est fourni par le protocole d'application correspondant faisant appel aux services proposés par la liaison de données ou une autre couche immédiatement inférieure. Le présent document définit les caractéristiques de service d'application que les applications de bus de terrain et/ou la gestion de systèmes peuvent exploiter.

Dans l'ensemble des normes de bus de terrain, le terme "service" fait référence à la capacité abstraite transmise par une couche du modèle de référence de base OSI à la couche immédiatement supérieure. Ainsi, le service de couche application défini dans le présent document est un service d'architecture conceptuel, indépendant des services d'administration et de mise en œuvre.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 5-20: Définition des services de la couche application – Eléments de type 20

1 Domaine d'application

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

La présente Norme internationale fournit des éléments communs pour les communications de messagerie de base, prioritaires et non prioritaires, entre les programmes d'application dans un environnement automatisé et les équipements spécifiques aux bus de terrain de Type 20. Le terme "prioritaire " est utilisé pour représenter la présence d'une fenêtre temporelle, dans le cadre de laquelle une ou plusieurs actions spécifiées sont tenues d'être réalisées avec un certain niveau de certitude défini. La non-réalisation des actions spécifiées dans le cadre de la fenêtre temporelle peut mener à l'échec des applications sollicitant les actions, avec un risque annexe pour les équipements, l'installation et éventuellement les personnes.

La présente Norme internationale définit de manière abstraite le service visible externe fourni par le Type 20 de Couche application de bus de terrain, en termes

- a) d'un modèle abstrait de définition des ressources (objets) d'application que les utilisateurs peuvent manipuler grâce au service FAL,
- b) des actions et événements primitifs du service;
- c) des paramètres associés à chaque action et à chaque événement primitifs, ainsi que la forme de ces derniers, et
- d) de l'interrelation entre ces actions et événements, et leurs séquences valides.

La présente Norme internationale a pour objet de définir les services fournis à l'utilisateur FAL à la limite entre l'utilisateur et la couche application du modèle de référence de bus de terrain.

La présente Norme internationale spécifie la structure et les services de la couche application de bus de terrain CEI, conformément au modèle de référence de base OSI (ISO/CEI 7498-1) et à la structure de la couche application OSI (ISO/CEI 9545).

Bien que ces services précisent, du point de vue des applications, de quelle manière sont émises et transmises la demande et les réponses, ils ne comportent aucune spécification indiquant ce que les applications de demande et de réponse vont faire à ces dernières. En fait, les aspects liés au comportement des applications ne sont pas spécifiés, et seule est précisée une définition du type de demandes et de réponses que ces applications peuvent transmettre/recevoir. Ceci permet aux utilisateurs FAL d'appliquer une plus grande souplesse dans la normalisation de ce type de comportement d'objet. Outre ces services, la présente Norme internationale définit également certains services d'appui destinés à fournir un accès à la FAL afin de contrôler certains aspects de son fonctionnement.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les

références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

NOTE Toutes les parties de la série CEI 61158, ainsi que la CEI 61784-1 et la CEI 61784-2 font l'objet d'une maintenance simultanée. Les références croisées à ces documents dans le texte se rapportent par conséquent aux éditions datées dans la présente liste de références normatives.

CEI 61158-1:2014, *Réseaux de communication industriels – Spécifications des bus de terrain – Partie 1: Présentation et lignes directrices des séries CEI 61158 et CEI 61784*

CEI 62591:2010, *Réseaux de communication industriels – Réseau de communications sans fil et profils de communication – WirelessHART™*

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

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

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

ISO/CEI 9545, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Structure de la couche Application*

ISO/CEI 10731, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base – Conventions pour la définition des services OSI*

ANSI/IEEE 754: *IEEE Standard for Floating-Point Arithmetic*

3 Termes, définitions, symboles, abréviations et conventions

Pour les besoins du présent document, les termes, définitions, abréviations, symboles et conventions suivants s'appliquent:

3.1 Termes et définitions d'autres normes ISO/CEI

3.1.1 Termes de l'ISO/CEI 7498-1

- a) syntaxe abstraite
- b) entité d'application
- c) processus d'application
- d) unité de données de protocole d'application
- e) élément de service d'application

3.1.2 Termes de l'ISO/CEI 9545

- a) invocation de l'entité d'application
- b) élément de service d'application
- c) élément de service d'application

3.1.3 Termes de l'ISO/CEI 8824-1

- a) identifiant d'objet
- b) type

- c) valeur
- d) type simple
- e) type structuré
- f) type de composant
- g) balise
- h) vrai
- i) faux
- j) type entier
- k) type de chaîne d'octets
- m) type nul

3.2 Termes de la CEI 61158-1

Pour les besoins du présent document, les termes et définitions suivants s'appliquent:

3.2.1

application

fonction ou structure de données pour laquelle des données sont consommées ou produites

3.2.2

objet d'application

classe d'objets qui gère et assure un échange de messages pendant le mode exécution à travers le réseau et à l'intérieur de l'appareil de réseau

Note 1 à l'article: De nombreux types de classes d'objets d'application peuvent être définis.

3.2.3

processus d'application

partie d'une application répartie sur un réseau, située sur un appareil et comportant une adresse non ambiguë

3.2.4

objet du processus d'application

composant d'un processus d'application identifiable et accessible par une relation entre applications FAL

Note 1 à l'article: Les définitions de l'objet du processus d'application sont constituées d'un ensemble de valeurs pour les attributs de leur classe (voir la définition de la Définition de classe d'objets du processus d'application). Les définitions de l'objet du processus d'application peuvent être accessibles à distance au moyen des services de l'ASE de gestion des objets FAL. Ces services peuvent être utilisés pour charger ou actualiser les définitions d'objets, pour lire les définitions d'objets et pour créer et supprimer de manière dynamique ces mêmes objets et leurs définitions correspondantes.

3.2.5

classe d'objets du processus d'application

classe des objets du processus d'application définis en termes d'ensemble de leurs attributs et services accessibles par le réseau

3.2.6

relation entre applications

association coopérative entre deux ou plusieurs invocations d'entités d'application (application-entity-invocation) à des fins d'échange d'informations et de coordination de leur fonctionnement conjoint

Note 1 à l'article: Cette relation est activée soit par l'échange d'unités de données de protocole d'application, soit à la suite d'activités de préconfiguration.

3.2.7**point d'extrémité de relation entre applications**

contexte et comportement d'une relation entre applications tels que vus et maintenus par l'un des processus d'application impliqués dans la relation entre applications

Note 1 à l'article: Chaque processus d'application impliqué dans la relation entre applications maintient son propre point d'extrémité de relation entre applications.

3.2.8**attribut**

description d'une caractéristique ou fonction visible de l'extérieur d'un objet

Note 1 à l'article: Les attributs d'un objet contiennent des informations relatives à des parties variables d'un objet. Typiquement, ils fournissent des informations de statut ou régissent le fonctionnement d'un objet. Des attributs peuvent aussi avoir une incidence sur le comportement d'un objet. Les attributs se répartissent en attributs de classes et attributs d'instances.

3.2.9**comportement**

indication de la façon dont l'objet réagit à des événements particuliers

Note 1 à l'article: Sa description inclut la relation entre les valeurs et services d'attributs.

3.2.10**classe**

ensemble d'objets, qui représentent tous le même type de composant système

Note 1 à l'article: Une classe est une généralisation de l'objet; un modèle pour définir des variables et des méthodes. Tous les objets dans une classe ont une forme et un comportement identiques, mais contiennent en général des données différentes dans leurs attributs.

3.2.11**attributs de classe**

attribut qui est partagé par tous les objets au sein de la même classe

3.2.12**code de classe**

identifiant unique attribué à chaque classe d'objets

3.2.13**service spécifique à la classe**

service défini par une classe d'objets particulière pour accomplir une fonction requise qui n'est pas accomplie par un service commun

Note 1 à l'article: Un objet spécifique à une classe est unique pour la classe d'objets qui le définit.

3.2.14**client**

a) objet qui utilise les services d'un autre objet (serveur) pour accomplir une tâche

b) initiateur d'un message auquel un serveur réagit, comme par exemple un point d'extrémité d'AR qui émet des APDU de demande de service confirmées vers un point d'extrémité d'AR unique agissant comme serveur

3.2.15**trajet d'acheminement**

flux unidirectionnel d'APDU dans le cadre d'une relation entre applications

3.2.16**cyclique**

terme utilisé pour décrire les événements qui se répètent de manière régulière et répétitive

3.2.17

extrémité (ou point d'extrémité)

une des entités de communication impliquées dans une connexion

3.2.18

erreur

discordance entre une valeur ou un état calculé(e), observé(e) ou mesuré(e) et la valeur ou l'état spécifié(e) ou théoriquement correct(e)

3.2.19

code d'erreur

identification d'un type spécifique d'erreur dans une classe d'erreurs

3.2.20

informations de gestion

informations accessibles sur le réseau qui prennent en charge la gestion du fonctionnement du système de bus de terrain, y compris la couche application

Note 1 à l'article: La gestion comprend des fonctions telles que le contrôle, la surveillance et le diagnostic.

3.2.21

serveur

a) rôle d'un AREP dans lequel il retourne une APDU de réponse de service confirmée au client qui a émis la demande

b) objet qui fournit des services à un autre objet (client)

3.2.22

service

opération ou fonction réalisée par un objet et/ou une classe d'objets sur demande d'un autre objet et/ou d'une autre classe d'objets

Note 1 à l'article: Un ensemble de services communs est défini et des dispositions sont prévues pour la définition des services spécifiques à l'objet. Les services spécifiques à l'objet sont ceux définis par une classe d'objets particulière afin d'exécuter une fonction requise non exécutée par un service commun.

3.3 Définitions spécifiques à la couche application de bus de terrain de type 20

Pour les besoins du présent document, les termes et définitions suivants s'appliquent:

3.3.1

voie analogique

signal électrique à variation continue permettant de relier un appareil de terrain au reste du système d'acquisition ou de contrôle des données

Note 1 à l'article: Certains appareils de terrain prennent en charge plusieurs voies analogiques (entrée ou sortie). Chaque voie analogique transmet une seule variable dynamique vers ou en provenance de l'appareil de terrain.

3.3.2

diffusion

processus de transmission d'une PDU (unité de données de protocole) à tous les appareils qui sont connectés au réseau et qui sont en mesure de recevoir la transmission

3.3.3

adresse de diffusion

adresse utilisée par un maître pour transmettre une commande à tous les appareils

3.3.4**en mode rafale**

initiation de l'activité de communication par un appareil esclave à intervalles cycliques sans demande d'un maître

3.3.5**erreur de communication**

erreur détectable lors de la réception d'une PhPDU ou DLPDU, également l'octet du "code d'erreur de communication" de l'APDU

3.3.6**appareil**

toute entité contenant une mise en œuvre de bus de terrain de Type 20

3.3.7**ID d'appareil**

numéro de série d'un appareil qui est unique parmi toutes les instances d'un type d'appareil

Note 1 à l'article: Le fabricant est tenu d'attribuer une valeur unique pour chaque appareil dont les valeurs sont identiques pour l'ID du fabricant et le type d'appareil.

3.3.8**type d'appareil**

type de fabrication d'un appareil, par exemple son nom de produit

Note 1 à l'article: La valeur de cet attribut est unique parmi tous les fabricants et tous les types d'appareils. Sa valeur spécifie l'ensemble des commandes et des objets de données pris en charge par l'appareil.

3.3.9**variable d'appareil**

élément de donnée à définition unique dans un appareil de terrain, qui est toujours associé aux informations relatives au processus

Note 1 à l'article: Une valeur de variable d'appareil varie suite à des modifications et des variations du processus auquel l'appareil est relié.

3.3.10**variable dynamique**

variable d'appareil qui est attribuée comme la valeur dynamique et éventuellement associée à une voie analogique

Note 1 à l'article: Un appareil peut contenir jusqu'à quatre variables - variables primaires, secondaires, tertiaires et quaternaires. Ces variables sont désignées collectivement par le terme "dynamiques".

3.3.11**type d'appareil étendu**

type de fabrication de l'appareil

Note 1 à l'article: La valeur de cet attribut est unique parmi tous les fabricants et tous les types d'appareils. Sa valeur spécifie l'ensemble des commandes et des objets de données pris en charge par l'appareil.

3.3.12**appareil de terrain**

entité physique connectée au processus ou à l'équipement et possédant au moins un élément de signalisation qui communique avec un ou des autre(s) élément(s) de signalisation via un câble

Note 1 à l'article: Il est directement connecté au capteur ou à la grille d'entraînement ou il exécute la fonction de commande de processus et est directement connecté à la couche physique spécifiée dans la présente norme. Il peut générer ou recevoir un signal analogique en plus d'un signal numérique.

3.3.13

balise longue

chaîne ISO Latin-1 restreinte à 32 caractères servant à identifier un appareil de terrain

3.3.14

courant de ligne

valeur mesurée par un mA en série avec l'appareil de terrain

Note 1 à l'article: Le courant de ligne est un signal 4 mA à 20 mA analogique en courant continu servant à communiquer une valeur unique entre le système de commande et l'appareil de terrain. Les appareils en mode tension utilisent "volts en courant continu" comme unités techniques lorsque les valeurs de "courant de ligne" sont utilisées.

3.3.15

ID du fabricant

chaîne identifiant le fabricant ayant produit l'appareil

Note 1 à l'article: Un fabricant est tenu d'utiliser la valeur qui lui est attribuée et n'est pas autorisé à utiliser la valeur attribuée à un autre fabricant.

3.3.16

maître

appareil qui initie une activité de communication en envoyant une trame de demande à un autre appareil et en attendant une trame de réponse de la part de cet appareil

3.3.17

réseau

paire unique de câbles, connecteurs, éléments de signalisation associés au moyen desquels un ensemble d'appareils de signalisation sont interconnectés, et les éléments ne servant pas à la signalisation qui sont raccordés à cette même paire de câbles

Note 1 à l'article: Une installation qui utilise plusieurs paires de fils et une alimentation de réseau commune est considérée comme plusieurs réseaux.

3.3.18

adresse d'interrogation

identifiant attribué à un appareil de manière unique dans le réseau auquel l'appareil est connecté

3.3.19

esclave

appareil qui initie une activité de communication uniquement après avoir reçu une trame de demande de la part d'un appareil maître et qui est tenu d'envoyer une réponse à cette demande

3.3.20

balise

chaîne ASCII de 8 caractères servant à identifier un appareil de terrain

3.3.21

ID unique

identifiant attribué à un appareil, unique parmi toutes les instances des appareils conformes à la présente norme

3.4 Abréviations et symboles

AE	Application Entity (Entité d'application)
AL	Application Layer (Couche application)
AP	Application Process (Processus d'application)

APDU	Application Protocol Data Unit (Unité de données de protocole d'application)
APO	Application Process Object (Objet de processus d'application)
AR	Application Relationship (Relation entre applications)
AREP	Application Relationship End Point (Point d'extrémité de relation entre applications)
ASCII	American Standard Code for Information Interchange Code américain normalisé pour l'échange d'informations)
ASE	Application Service Element (Elément de service d'application)
Cnf	Confirmation
DL	Data Link (Liaison de données) (comme préfixe)
DLC	Data Link Connection (Connexion de liaison de données)
DLL	Data Link Layer (Couche liaison de données)
DLM	Data Link-management (Gestion de liaison de données)
DLSAP	Data Link Service Access Point (Point d'accès au service de liaison de données)
DLSDU	DL-service-data-unit (Unité de données de service DL)
FAL	Fieldbus Application Layer (Couche application de bus de terrain)
HCF	HART [®] Communication Foundation
ID	Identifiant
CEI	Commission électrotechnique internationale
Ind	Indication
OSI	Open Systems Interconnect (Interconnexion de systèmes ouverts)
Req	Demande
Rsp	Réponse
VFD	Virtual Field Device (Appareil de terrain virtuel)

3.5 Conventions

3.5.1 Vue d'ensemble

La FAL est définie comme un ensemble d'ASE orientés objet. Chaque ASE est spécifié dans un paragraphe séparé. Chaque spécification d'ASE est constituée de deux parties, à savoir sa spécification de classe et sa spécification de service.

La spécification de classe définit les attributs de la classe. La spécification de service définit les services fournis par l'ASE.

3.5.2 Conventions pour les définitions de classe

Les définitions de classe sont décrites au moyen de modèles. Chaque modèle consiste en une liste d'attributs propres à la classe. La forme générale du modèle est présentée ci-dessous:

FAL ASE:	Nom de ASE
CLASS:	Nom de classe
CLASS ID:	#
PARENT CLASS:	Nom de la classe parent
ATTRIBUTES:	
1 (o) Attribut clé:	identifiant numérique
2 (o) Attribut clé:	nom
3 (m) Attribut:	nom de l'attribut(valeurs)

4	(m)	Attribut:	nom de l'attribut(valeurs)
4.1	(s)	Attribut:	nom de l'attribut(valeurs)
4.2	(s)	Attribut:	nom de l'attribut(valeurs)
4.3	(s)	Attribut:	nom de l'attribut(valeurs)
5	(c)	Contrainte:	expression de contrainte
5.1	(m)	Attribut:	nom de l'attribut(valeurs)
5.2	(o)	Attribut:	nom de l'attribut(valeurs)
6	(m)	Attribut:	nom de l'attribut(valeurs)
6.1	(s)	Attribut:	nom de l'attribut(valeurs)
6.2	(s)	Attribut:	nom de l'attribut(valeurs)

SERVICES:

1	(o)	OpsService:	nom de service
2	(c)	Contrainte:	expression de contrainte
2.1	(o)	OpsService:	nom de service
3	(m)	MgtService:	nom de service

- (1) L'entrée "FAL ASE:" désigne le nom de l'ASE FAL qui fournit les services pour la classe spécifiée.
- (2) L'entrée "CLASS" désigne le nom de la classe spécifiée. Tous les objets définis au moyen de ce modèle constituent une instance de cette classe. La classe peut être spécifiée par la présente norme ou par un utilisateur de cette dernière.
- (3) L'entrée "CLASS ID" désigne un nombre qui identifie la classe spécifiée. Ce nombre est unique au sein de la ASE FAL qui fournit les services pour cette classe. Lorsqu'il est qualifié par l'identité de son ASE FAL, ce nombre identifie de manière non ambiguë la classe dans le domaine d'application de la FAL. La valeur "NULL" indique que la classe ne peut pas être instanciée. La "CLASS ID" n'est pas utilisée dans le présent document.
- (4) L'entrée "PARENT CLASS" désigne le nom de la classe parent pour la classe spécifiée. Tous les attributs définis pour la classe parent et hérités par cette dernière sont également hérités pour la classe définie, et ne sont donc pas à redéfinir dans le modèle utilisé pour cette classe.

NOTE La classe parent "TOP" indique que la classe définie correspond à une définition de la classe initiale. La classe parent "TOP" est utilisée comme point de départ qui permet de définir toutes les autres classes. L'utilisation de l'entité "TOP" est réservée aux classes définies par le présent document..

- (5) L'étiquette "ATTRIBUTES" indique que les entrées suivantes sont des attributs définis pour la classe.
 - a) Chacune des entrées d'attributs contient un numéro de ligne dans la colonne 1, un indicateur obligatoire (m) / facultatif (o) / conditionnel (c) / sélectif (s) dans la colonne 2, une balise de type attribut dans la colonne 3, un nom ou une expression conditionnelle dans la colonne 4 et, le cas échéant, une liste des valeurs énumérées dans la colonne 5. La valeur par défaut de l'attribut peut être spécifiée dans la colonne suivant la liste des valeurs.
 - b) Les objets sont identifiés normalement par un identifiant numérique ou par un nom d'objet, ou par les deux. Dans les modèles de classe, ces attributs clés sont définis sous l'attribut clé
 - c) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication est utilisée pour spécifier
 - i) les champs d'un attribut structuré (4.1, 4.2, 4.3),
 - ii) les attributs soumis à une déclaration de contrainte (5). Les attributs peuvent être obligatoires (5.1) ou facultatifs (5.2) si la contrainte est vraie. Les attributs facultatifs n'exigent pas tous des déclarations de contrainte contrairement à l'attribut défini en (5.2).
 - iii) les champs de sélection d'un attribut du type de choix (6.1 et 6.2).
- (6) L'étiquette "SERVICES" indique que les entrées suivantes sont des services définis pour la classe.

- a) Un (m) dans la colonne 2 indique que le service est obligatoire pour la classe, tandis qu'un (o) indique que ce dernier est facultatif. Un (c) dans cette colonne indique que le service est conditionnel. Lorsque tous les services définis pour une classe sont définis comme facultatifs, au moins un de ces services est à sélectionner lorsqu'une instance de la classe est définie.
- b) L'étiquette "OpsService" désigne un service opérationnel.
- c) L'étiquette "MgtService" désigne un service de gestion.
- d) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication dans la liste des services permet de spécifier les services soumis à une déclaration de contrainte

3.5.3 Conventions pour les définitions de service

3.5.3.1 Généralités

Le présent document utilise les conventions de description données dans l'ISO/CEI 10731.

Le modèle de service, les primitives de service et les diagrammes séquentiels de temps utilisés sont des descriptions entièrement abstraites; ils ne représentent pas une spécification de mise en œuvre.

3.5.3.2 Paramètres de services

Les primitives de service sont utilisées pour représenter les interactions entre utilisateur de service et fournisseur de service (ISO/CEI 10731). Elles acheminent des paramètres qui indiquent les informations disponibles dans l'interaction utilisateur/fournisseur.

Le présent document utilise un format tabulaire pour décrire les paramètres de composants des primitives de service. Les paramètres qui s'appliquent à chaque groupe de primitives de service sont définis dans des tableaux jusqu'à la fin du présent document. Chaque tableau comprend jusqu'à six colonnes: une colonne pour le nom du paramètre de service, ainsi qu'une colonne pour chacune de ces primitives et le sens de transfert des paramètres utilisés par le service. Les six colonnes possibles sont les suivantes:

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

NOTE 1 Cette capacité est rarement utilisée. Sauf spécification contraire, les paramètres de primitive de demande sont des paramètres d'entrée.

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

NOTE 2 Les primitives de demande, indication, réponse et confirmation sont également connues comme primitives requestor.submit (soumission par le demandeur), accepter.deliver (remise à l'accepteur), accepter.submit (soumission par l'accepteur) et requestor.deliver (remise au demandeur), respectivement (voir ISO/CEI 10731).

Un paramètre (ou composant) est présenté dans chaque rangée de chaque tableau. Dans les colonnes des primitives de service concernées, un code permet de spécifier le type d'utilisation du paramètre sur la primitive spécifiée dans la colonne:

- M le paramètre est obligatoire pour la primitive
- U le paramètre est une option de l'utilisateur, et peut ou non être fourni selon l'utilisation dynamique de l'utilisateur du service. Lorsque ce paramètre n'est pas fourni, une valeur par défaut y afférent est supposée.
- C le paramètre dépend d'autres paramètres ou de l'environnement de l'utilisateur du service.

— (blanc/vide) le paramètre n'est jamais présent.

S le paramètre est un élément sélectionné.

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

- a) une contrainte spécifique au paramètre:
“(=)” indique que le paramètre est sémantiquement équivalent au paramètre dans la primitive de service, par rapport à son équivalent gauche immédiat dans le tableau.
- b) une indication stipulant qu'une note s'applique à l'entrée:
“(n)” indique que la note suivante "n" contient des informations supplémentaires relatives au paramètre et à son utilisation.

3.5.3.3 Procédures de service

Les procédures sont définies en termes

- d'interactions entre les entités d'application par l'échange des unités de données de protocole d'application de bus de terrain, et
- d'interactions entre un fournisseur de service de couche application et un utilisateur de service de couche application dans le même système par l'invocation des primitives de service de couche application.

Ces procédures s'appliquent aux instances de communication entre des systèmes qui prennent en charge des services de communications prioritaires dans la couche application de bus de terrain.

NOTE La sous-série de normes CEI 61158-5 définit des ensembles de services abstraits. Il ne s'agit ni de spécifications de protocole, ni de spécifications de mise en œuvre, et ni de spécifications d'interface de programmation concrètes. Des restrictions s'appliquent par conséquent sur le degré d'autorisation potentiel des procédures de service dans les parties de la sous-série CEI 61158-5. Les aspects de protocole pouvant varier parmi les différentes spécifications de protocole ou les différentes mises en œuvre d'instanciation des mêmes services abstraits ne sont pas adaptés à une intégration dans ces définitions de service, sauf au niveau d'abstraction nécessairement commun à l'ensemble de ces expressions.

Par exemple, la méthode qu'utilisent les fournisseurs de service pour l'appariement des PDU de demande et de réponse est appropriée à la spécification dans un document de spécification de protocole de la sous-série CEI 61158-6, mais non dans un document de définition de services abstraits de la sous-série CEI 61158-5. De façon similaire, les méthodes de mise en œuvre locales qu'utilise un fournisseur ou un utilisateur de service pour apparier les primitives de demande et de confirmation, ou les primitives d'indication et de réponse, sont adaptées à une spécification de mise en œuvre ou à une spécification d'interface de programmation, mais non à un document de service abstrait ou à un document de protocole, sauf à un niveau d'abstraction nécessairement commun à toutes les représentations du document de spécification. Dans tous les cas, il n'est pas admis que la définition abstraite spécifie outre mesure la réalisation d'instanciation plus concrète.

Des informations supplémentaires concernant les procédures de service conceptuel d'une mise en œuvre d'un protocole d'application des services de l'une des définitions de services abstraits de la sous-série CEI 61158-5, figurent en 9.6 de la CEI 61158-1.

4 Concepts

Les concepts et modèles communs utilisés pour décrire le service de la couche application dans le présent document sont détaillés à l'Article 9 de la CEI 61158-1.

5 ASE des types de données

5.1 Vue d'ensemble

5.1.1 Généralités

Les types de données de bus de terrain spécifient la syntaxe transportable pour les données d'application acheminées par les services FAL. La couche application de bus de terrain prend

en charge la définition et le transfert des types de données de base et construits. Les règles de codage applicables aux types de données spécifiés dans le présent article sont mentionnées dans la sous-série CEI 61158-6.

Les types de base sont des types atomiques qui ne peuvent pas être décomposés en types plus élémentaires. Les types construits sont des types composés de types de base et d'autres types construits. Le présent document n'impose pas de contrainte sur leur complexité et leur intensité d'imbrication.

Les types de données sont définis comme des instances de la classe de type de données, tel qu'illustré à la Figure 1.

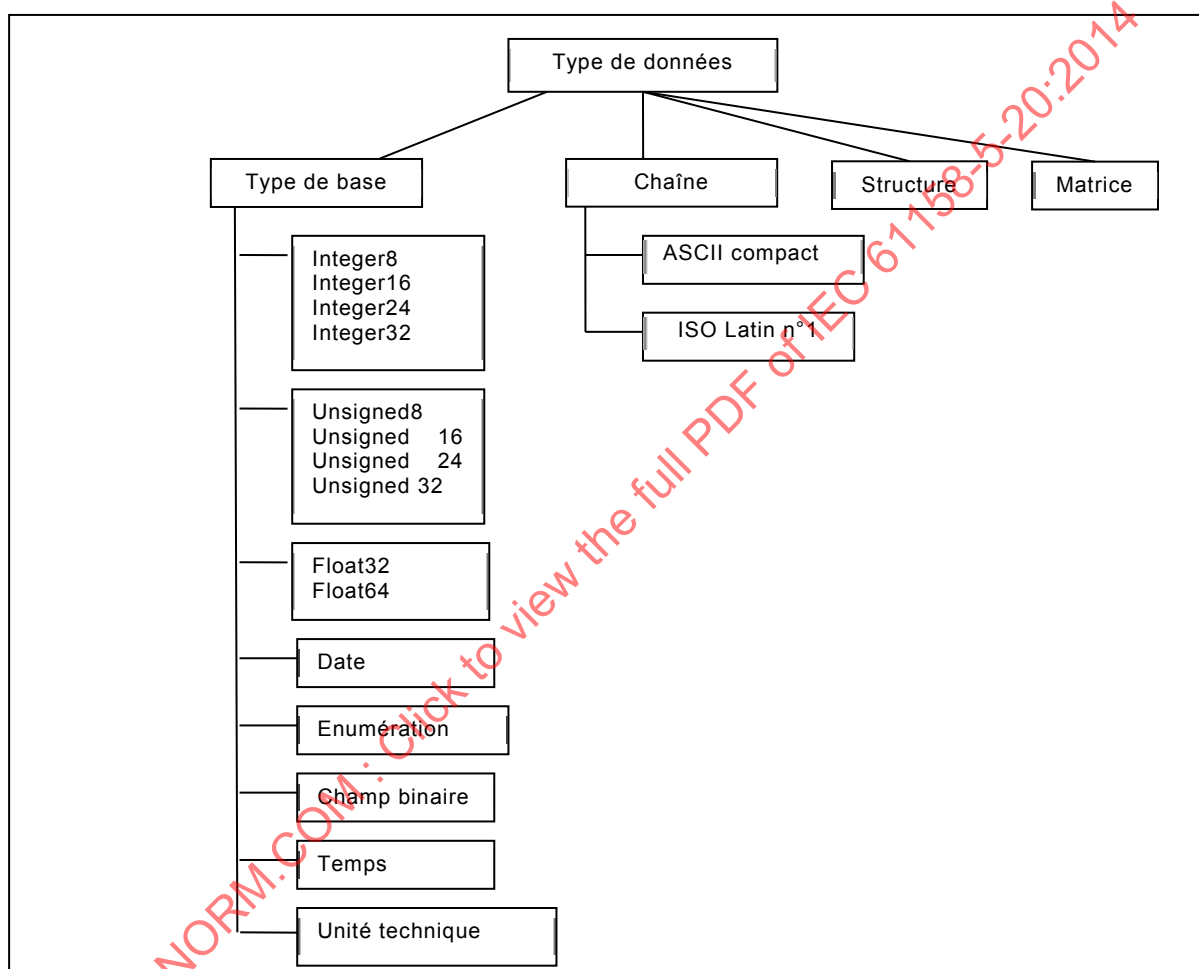


Figure 1 – Hiérarchie des classes de types de données

Les définitions des types de données sont représentées comme une structure classe/format/instance commençant par la classe de type de données intitulée "Type de données". Les formats de types de données sont définis par la classe y afférent.

Les classes de données de base sont toujours des types de données de longueur fixe. Les types communs issus de l'ISO/CEI 8824-1 sont désignés comme des types de données *simples*. D'autres types de données de base communs sont définis spécifiquement pour les applications de bus de terrain et sont désignés comme des *types spécifiques*.

Les types construits spécifiés dans le présent document sont des chaînes, des matrices et des structures. Aucun type commun n'est défini pour les matrices et les structures.

5.1.2 Types de base

La plupart des types de base sont définis à partir d'un ensemble de types définis dans l'ISO/CEI 8824-1 (types simples). Certains types définis dans l'ISO/CEI 8824-1 ont été étendus pour une utilisation spécifique aux bus de terrain (types spécifiques).

Les types simples sont des types universels définis dans l'ISO/CEI 8824-1. Ils sont définis dans le présent document de manière à comporter des identifiants de classe de bus de terrain.

Les types spécifiques sont des types de base définis spécifiquement pour une utilisation dans l'environnement de bus de terrain. Ils sont définis comme des sous-types de classes simples.

Les types de base ont une longueur constante. Deux variations sont définies, à savoir une pour définir les types de données dont la longueur est un entier relatif d'octets, et une pour définir les types de données dont la longueur correspond à des bits.

NOTE Le présent document définit les éléments Entier, ASCII compact, ISO Latin n°1 et Date afin de leur attribuer des identifiants de classe de bus de terrain. Le présent document ne modifie pas leurs définitions spécifiées dans l'ISO/CEI 8824-1.

5.1.3 Types construits

5.1.3.1 Vue d'ensemble

Les types de données construits sont nécessaires pour l'acheminement complet des diverses informations présentes sur le bus de terrain. Il existe trois types de données construits définis pour le présent document, à savoir les chaînes, les matrices et les structures.

5.1.3.2 Chaîne

Une chaîne est composée d'un ensemble ordonné, de nombre variable, d'éléments de longueur fixe et de type homogène.

5.1.3.3 Structure

Une structure est constituée d'un ensemble ordonné d'éléments de type hétérogène dénommés champs. Le présent document ne restreint pas le type de données des champs. Cependant, les champs d'une structure n'ont pas à être de même type.

5.1.3.4 Matrice

Une matrice est composée d'un ensemble ordonné d'éléments de type homogène. Le type de données des éléments d'une matrice peut être un type de base ou une structure à longueur fixe. Tous les éléments d'une matrice doivent être du même type de données.

5.1.3.5 Niveau d'imbrication

Le présent document permet aux structures et aux matrices de contenir des structures et des matrices.

5.1.4 Spécification des types de données définis par l'utilisateur

Les utilisateurs peuvent considérer comme nécessaire de définir des types de données personnalisés pour leurs propres applications. Le présent document ne prend pas en charge les types définis par l'utilisateur.

5.1.5 Transfert des données de l'utilisateur

Les données de l'utilisateur sont transférées entre les applications par le protocole FAL. Toutes les opérations de codage et décodage sont effectuées par l'utilisateur FAL.

Les règles de codage des données de l'utilisateur en unités de données de protocole FAL sont dépendantes du type de données. Ces règles sont définies dans la CEI 61158-6-20. Les types de données définis par l'utilisateur, pour lesquels il n'existe aucune règle de codage, sont transférés sous forme de séquence d'octets de longueur variable. Le format des données dans la chaîne d'octets est défini par l'utilisateur.

5.2 Définition formelle des objets de types de données

5.2.1 Classe de type de données

La classe de type de données spécifie la racine de l'arborescence des classes de types de données. Sa classe parent "top" indique le sommet de l'arborescence des classes FAL.

FAL ASE: DATA TYPE ASE

CLASS: DATA TYPE

CLASS ID: Non utilisé

PARENT CLASS: TOP

ATTRIBUTES:

- | | | | |
|-------|-----|---------------|---|
| 1 | (m) | Attribut clé: | Nom de type de données |
| 2 | (m) | Attribut: | Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY) |
| 3 | (c) | Contrainte | Format = FIXED LENGTH STRING |
| 3.1 | (m) | Attribut: | Longueur en octets |
| 4 | (c) | Contrainte: | Format = STRUCTURE |
| 4.1 | (m) | Attribut: | Nombre de champs |
| 4.2 | (m) | Attribut: | Liste de champs |
| 4.2.1 | (o) | Attribut: | Nom de champ |
| 4.2.2 | (m) | Attribut: | Type de données de champ |
| 5 | (c) | Contrainte: | Format = ARRAY |
| 5.1 | (m) | Attribut: | Nombre d'éléments de matrice |
| 5.2 | (m) | Attribut: | Type de données d'éléments de matrice |

5.2.2 Attributs

Format

Cet attribut identifie le type de données comme une structure à longueur fixe, une chaîne, une matrice ou une structure de données.

Longueur en octets

Cet attribut conditionnel définit la représentation des dimensions de l'objet de type associé. Il est présent lorsque la valeur de l'attribut de format est "FIXED LENGTH" ou "STRING". Pour les types de données FIXED LENGTH, il représente la longueur en octets. Pour les types de données STRING, il représente la longueur en octets pour un élément unique d'une chaîne.

Nombre de champs

Cet attribut conditionnel définit le nombre de champs dans une structure. Il est présent lorsque la valeur de l'attribut de format est "STRUCTURE".

Liste de champs

Cet attribut conditionnel est une liste ordonnée de champs contenus dans la structure. Chaque champ est spécifié par son numéro et son type. Les champs sont numérotés de manière séquentielle à partir de 0 (zéro) dans leur ordre d'apparition. L'accès partiel aux champs d'une structure n'est pas pris en charge.

Nom de champ

Cet attribut facultatif et conditionnel spécifie le nom du champ. Il peut être présent lorsque la valeur de l'attribut de format est "STRUCTURE".

Type de données de champ

Cet attribut conditionnel spécifie le type de données du champ. Il est présent lorsque la valeur de l'attribut de format est "STRUCTURE". Cet attribut peut spécifier de lui-même un type de données construit soit par référencement de la définition d'un type de données construit soit par intégration de la définition d'un type de données construit à ce niveau.

Nombre d'éléments de matrice

Cet attribut conditionnel définit le nombre d'éléments pour le type de matrice. Les éléments de matrice sont indexés en commençant par "0" jusqu'à "n-1", lorsque la taille de la matrice correspond à "n" éléments. Cet attribut est présent lorsque la valeur de l'attribut de format est "ARRAY".

Type de données d'éléments de matrice

Cet attribut conditionnel spécifie le type de données pour les éléments d'une matrice. Tous les éléments de matrice ont le même type de données. Il est présent lorsque la valeur de l'attribut de format est "ARRAY". Cet attribut peut spécifier de lui-même un type de données construit en référençant un type de données construit par son nom.

5.3 Types de données définis par la FAL

5.3.1 Types de longueur fixe

5.3.1.1 Integer8

CLASS:	Data type (Type de données)
ATTRIBUTES:	
1 Nom de type de données	= Integer8
2 Format	= FIXED LENGTH
2.1 Longueur en octets	= 1

Ce type d'entier est un double nombre binaire complémentaire avec une longueur de un octet.

5.3.1.2 Integer16

CLASS:	Data type
ATTRIBUTES:	
1 Nom de type de données	= Integer16
2 Format	= FIXED LENGTH
2.1 Longueur en octets	= 2

Ce type d'entier est un double nombre binaire complémentaire avec une longueur de deux octets.

5.3.1.3 Integer24

CLASS:	Data type
ATTRIBUTES:	
1 Nom de type de données	= Integer24
2 Format	= FIXED LENGTH
2.1 Longueur en octets	= 3

Ce type d'entier est un double nombre binaire complémentaire avec une longueur de trois octets.

5.3.1.4 Integer32

CLASS:	Data type
--------	-----------

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Integer32 |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 4 |

Ce type d'entier est un double nombre binaire complémentaire avec une longueur de quatre octets.

5.3.1.5 Unsigned8

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Unsigned8 |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 1 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet correspondant est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type a une longueur de un octet.

5.3.1.6 Unsigned16

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Unsigned16 |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 2 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet correspondant est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type non signé a une longueur de deux octets.

5.3.1.7 Unsigned24

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Unsigned24 |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 3 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet correspondant est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type non signé a une longueur de trois octets.

5.3.1.8 Unsigned32

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Unsigned32 |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 4 |

Ce type est un nombre binaire. Le bit de poids fort de l'octet correspondant est toujours utilisé comme le bit de poids fort du nombre binaire; aucun bit de signe n'est inclus. Ce type non signé a une longueur de quatre octets.

5.3.1.9 Float32

CLASS: Data type

ATTRIBUTES:

- | | | | |
|---|------------------------|---|---------|
| 1 | Nom de type de données | = | Float32 |
|---|------------------------|---|---------|

- | | | | |
|-----|--------------------|---|--------------|
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 4 |

Ce type a une longueur de quatre octets. Le format de Float32 est celui défini par l'ANSI/IEEE 754 comme un format de simple précision.

5.3.1.10 Float64

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Float64 |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 8 |

Ce type a une longueur de huit octets. Le format de Float64 est celui défini par l'ANSI/IEEE 754 comme un format de double précision.

5.3.1.11 Date

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Date |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 3 |

Ce type de données est constitué du jour, du mois et de l'année moins 1900. Ceci permet la représentation de toute date comprise entre le 1er janvier 1900 et le 31 décembre 2155.

5.3.1.12 Enumération

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Enumération |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 1 |

Les éléments de données à signification unique provenant d'une liste ou d'un tableau sont codés sous le format Enumération. Ce type de données utilise un entier non signé d'une longueur de un octet. La valeur entière la plus grande est réservée et n'est utilisée par aucun service.

5.3.1.13 Champ binaire

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|---------------|
| 1 | Nom de type de données | = | Champ binaire |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 1 |

Ce type de données est défini comme une série de huit bits, numérotés de 0 à 7. La communication des informations codées comme des données à un seul bit (telles que les informations d'état et de diagnostic) utilise ce type de données.

5.3.1.14 Temps

CLASS: Data type

ATTRIBUTES:

- | | | | |
|-----|------------------------|---|--------------|
| 1 | Nom de type de données | = | Temps |
| 2 | Format | = | FIXED LENGTH |
| 2.1 | Longueur en octets | = | 4 |

Ce type de données est un entier binaire non signé et représente l'heure dans les incréments de $1/32^{\text{ème}}$ de milliseconde. Si ce type de données est utilisé pour représenter l'heure du jour, il indique donc le nombre $1/32^{\text{ème}}$ de millisecondes depuis minuit.

5.3.1.15 Unité technique

CLASS: Data type

ATTRIBUTES:

- 1 Nom de type de données = Unité technique
- 2 Format = FIXED LENGTH
- 2.1 Longueur en octets = 1

Ce type définit l'unité de mesure d'une variable mesurée. L'interprétation de ce type de données est spécifiée par le profil de communication et au-delà du domaine d'application de la présente partie du document.

5.3.2 Types de chaînes

5.3.2.1 ASCII compact

CLASS: Data type

ATTRIBUTES:

- 1 Nom de type de données = ASCII compact
- 2 Format = STRING
- 2.1 Longueur en octets = 1 à n

Ce type est un sous-ensemble modifié de l'ensemble de codes de caractères ASCII. Ce sous-ensemble est présenté dans le Tableau 1.

Tableau 1 – Ensemble de caractères ASCII compact

Bits 4 et 5	Bits 0 à 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
2	SP ^a	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
NOTE Le chiffre hexadécimal de poids fort se lit de haut en bas; le chiffre de poids faible se lit de gauche à droite.																
^a SP indique un caractère espace.																

5.3.2.2 ISO Latin n°1

CLASS: Data type

ATTRIBUTES:

- 1 Nom de type de données = Latin n°1
- 2 Format = STRING
- 2.1 Longueur en octets = 1 à n

Ce type est un sous-ensemble du type de chaîne de l'ISO/CEI 8859-1 (ISO latin n°1). Ce sous-ensemble est présenté dans le Tableau 2.

Tableau 2 – Caractères ISO Latin n°1

Bits 4 à 7	Bits 0 à 3															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0																
1																
2	SP ^a	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	
8																
9																
A	NBSP ^b	ı	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	SHY ^c	®	¯
B	°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

NOTE Les chiffres hexadécimaux de poids fort sont classés de haut en bas; les chiffres hexadécimaux de poids faible sont classés de gauche à droite. Les cellules grisées indiquent qu'aucun caractère n'est attribué à ce code.

^a SP indique un caractère espace.

^b NBSP indique un caractère espace continu.

^c SHY indique un tiret.

5.4 Spécification des services ASE pour le type de données

Aucun service opérationnel n'est défini pour l'objet de type.

5.5 Synthèse des types de données

Le présent article contient une synthèse des types de données définis tel que présenté dans le Tableau 3.

Tableau 3 – Synthèse des types de données

Type de données	Article	Type de données	Article
Integer8	5.3.1.1	Float32	5.3.1.9
Integer16	5.3.1.2	Float64	5.3.1.10
Integer24	5.3.1.3	Date	5.3.1.11
Integer32	5.3.1.4	Enumération	5.3.1.12
Unsigned8	5.3.1.5	Champ binaire	5.3.1.13
Unsigned16	5.3.1.6	Temps	5.3.1.15
Unsigned24	5.3.1.7	Unité technique	5.3.1.15
Unsigned32	5.3.1.8	ASCII compact	5.3.2.1
		ISO Latin n°1	5.3.2.2

6 Spécification du modèle de Communication

6.1 Paramètres communs

Plusieurs paramètres sont utilisés par plus d'un service. Les définitions communes suivantes sont fournies en lieu et place d'une définition pour chaque service.

6.1.1 AREP ID

Ce paramètre spécifie des informations suffisantes pour identifier l'AREP de l'extrémité distante de l'AR. Une valeur de ce paramètre est réservée comme adresse de diffusion.

6.1.2 Code de réponse

En l'absence d'erreur de communication, ce paramètre spécifie un rapport d'exécution de commande qui indique l'état de l'exécution de la commande par l'appareil. Les valeurs possibles de ce paramètre sont présentées dans le Tableau 4. L'Énumération est son type de données.

Tableau 4 – Valeurs de code de réponse

Valeur	Description
Réussite	Exécution correcte de la commande (lecture ou écriture).
Avertissement	Exécution de la commande (écriture) avec l'écart décrit dans la réponse (par exemple, une valeur a été réglée sur sa valeur légale la plus proche).
Erreur	Exécution non correcte de la commande (lecture ou écriture). Le code de réponse indique la raison (par exemple; l'appareil est en mode protection d'écriture.)

6.1.3 Etat du processus d'application

Ce paramètre indique l'état fourni par l'utilisateur de la FAL et il n'est pas associé à l'exécution d'une commande donnée. Sa longueur est de un octet.

6.1.4 Etat étendu

Ce paramètre indique l'état du VFD. Il n'est pas associé à l'exécution d'une commande donnée. Sa longueur est de un octet.

6.1.5 Compte des préambules

Ce paramètre spécifie le nombre minimum d'octets de préambule contenus dans la DLPDU requise par la DLE de destination.

NOTE La valeur de ce paramètre peut être obtenue à l'aide du service "Identification" de la Couche application.

6.1.6 Etat de communication

Ce paramètre spécifie les informations concernant la défaillance de communication. Les valeurs possibles de ce paramètre sont présentées dans le Tableau 5.

Tableau 5 – Valeurs d'état de communication

Valeur	Description
Erreur de parité transversale	La parité d'un ou de plusieurs octets reçus par l'appareil n'était pas paire.
Erreur de dépassement	Au moins un octet des données du tampon de réception du PhE a été écrasé avant d'être lu (c'est-à-dire que le récepteur n'a pas traité l'octet entrant suffisamment rapidement).
Erreur de tramage	Le bit d'arrêt d'un ou de plusieurs octets reçus par l'appareil n'a pas été détecté par le PhE (c'est-à-dire qu'un repère ou la valeur 1 n'a pas été détecté(e) lorsqu'il convient qu'il se soit produit).
Erreur de parité longitudinale	La parité longitudinale calculée par l'appareil n'était pas adaptée à l'octet de contrôle à la fin de la DLPDU.
Dépassement de capacité du tampon	La PhPDU ou la DLPDU était trop longue pour le tampon de réception du PhE ou de la DLE
Appareil non disponible	Le client n'a reçu aucune réponse du serveur.

6.2 ASE

6.2.1 ASE d'appareil de terrain virtuel

6.2.1.1 Spécification des classes d'appareil de terrain virtuel

L'appareil de terrain virtuel (VFD) est un modèle abstrait de description des données et du comportement d'un processus d'application. Les VFD contiennent des APO. Les attributs d'un APO sont décrits par les descriptions d'objets. Les services sont définis pour un accès aux APO des VFD, tel que présenté à la Figure 2.

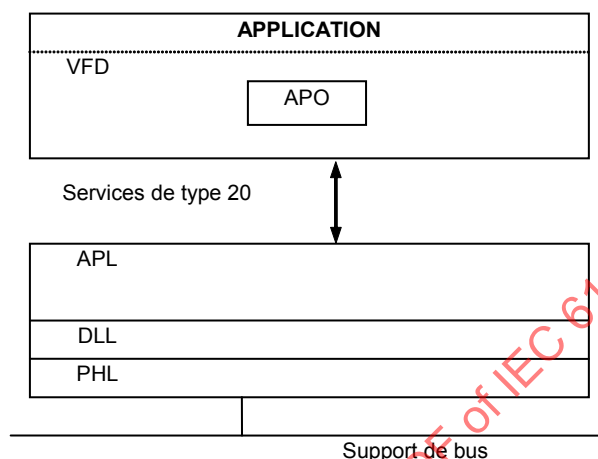


Figure 2 – Modèle VFD

Les services ne définissent aucune interface concrète de mise en œuvre. Ils décrivent sous une forme abstraite les fonctions pouvant être utilisées.

L'application n'est pas l'objet du présent document. Il convient d'indiquer uniquement de quelle manière les services décrits de façon abstraite peuvent être mis à disposition de l'application.

Un appareil ne comporte qu'un seul objet VFD.

6.2.1.2 Modèle formel

La classe VFD spécifie les attributs et services définis pour les processus d'application. Sa classe parent "top" indique le sommet de l'arborescence des classes.

ASE:	VFD ASE
CLASS:	VFD
CLASS ID:	—
PARENT CLASS:	TOP
1. (o) Attribut clé:	Non utilisé
2. (m) Attribut:	ID du fabricant
3. (m) Attribut:	Type d'appareil étendu
4. (m) Attribut:	ID d'appareil
5. (m) Attribut:	Rév. d'appareil
6. (m) Attribut:	Rév. logicielle
7. (m) Attribut:	Rév. matérielle
8. (m) Attribut:	Type phy
9. (m) Attribut:	Compte des préambules
10. (m) Attribut:	Drapeau d'appareil
11. (m) Attribut:	Rév. de commande

12. (m) Attribut:	Compte de variables
13. (m) Attribut:	Compteur de modification de configuration
14. (m) Attribut:	ExtdStatus d'appareil
15. (m) Attribut:	Code de diffusion
16. (m) Attribut:	Profil d'appareil

SERVICES:

1. (m) Ops Service: Identification

6.2.1.2.1 Attributs**ID du fabricant**

Cet attribut désigne le fabricant ayant produit l'appareil. Un fabricant doit utiliser la valeur qui lui est attribuée et il n'est pas admis qu'il utilise la valeur attribuée à un autre fabricant.

Type d'appareil étendu

Cet attribut indique le type de fabrication de l'appareil, c'est-à-dire le nom du produit. La valeur de cet attribut est attribuée par le fabricant. Sa valeur spécifie l'ensemble des commandes et des objets de données pris en charge par l'appareil. Le fabricant est tenu d'attribuer une valeur unique à chaque type d'appareil.

ID d'appareil

Cet attribut indique un numéro de série de l'appareil. Le fabricant est tenu d'attribuer une valeur unique pour chaque appareil dont les valeurs sont identiques pour l'ID du fabricant et le type d'appareil.

Rév d'appareil

Cet attribut décrit le niveau de révision de l'appareil. La valeur de cet attribut est définie par le fabricant. La valeur de cet attribut décrit le niveau de révision de l'ensemble des commandes et des objets de données pris en charge par l'appareil.

Rév logicielle

Cet attribut décrit le niveau de révision du micrologiciel de l'appareil. Le fabricant est tenu d'augmenter la valeur de cet attribut pour chaque nouvelle version du micrologiciel de l'appareil.

Rév matérielle

Cet attribut indique le niveau de révision du matériel de l'appareil. Le fabricant est tenu d'augmenter la valeur de cet attribut pour chaque modification importante du matériel de l'appareil. La traçabilité des modifications de composants matériels individuels n'est pas nécessaire.

Type phy

Cet attribut indique le type d'émission de signaux de la couche physique utilisée par l'appareil.

Compte des préambules

Cet attribut indique le nombre minimum de préambules à transmettre avec le message de demande entre le maître et l'appareil esclave.

Drapeau d'appareil

Cet attribut indique d'autres informations concernant l'appareil, telles que appareil à capteurs multiples, commande de mémoire non volatile, passerelle de protocole, etc.

Rév de Commande

Cet attribut indique le niveau de révision principal du protocole pris en charge par l'appareil.

Compte de variables

Cet attribut spécifie le nombre maximum d'objets (variables) accessibles depuis l'appareil. La valeur de cet attribut indique le dernier code de variable qu'une application client peut s'attendre à trouver dans l'appareil.

Compteur de modification de configuration

Cet attribut comptabilise le nombre de modifications de configuration de l'appareil. L'appareil est tenu d'augmenter la valeur de cet attribut à chaque réception d'une demande de modification de la configuration au moyen des services de la couche application, ou à chaque fois qu'un utilisateur de l'appareil modifie la configuration de l'appareil en appliquant des moyens locaux tels que l'interface locale de l'opérateur.

ExtdStatus d'appareil

Cet attribut indique l'état opérationnel étendu de l'appareil.

Code de diffusion

Cet attribut désigne le fabricant de marque privé ayant diffusé l'appareil.

Profil d'appareil

Cet attribut spécifie la classe d'appartenance de l'appareil.

6.2.1.3 Service d'identification

6.2.1.3.1 Utilisation

Ce service permet de demander à l'AP les informations d'identification de l'appareil.

6.2.1.3.2 Primitives de service

Les paramètres de service applicables à ce service sont présentés dans le Tableau 6.

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2014

Tableau 6 – Paramètres de service d'identification

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Balise	C,U	M (=)		
Résultat (+)			S	S (=)
ID du fabricant			M	M (=)
Type d'appareil étendu			M	M (=)
ID d'appareil			M	M (=)
Rév. d'appareil			M	M (=)
Rév. logicielle			M	M (=)
Rév. matérielle			M	M (=)
Type phy			M	M (=)
Compte des préambules			M	M (=)
Drapeau d'appareil			M	M (=)
Rév. de commande			M	M (=)
Compte de variables			M	M (=)
Compteur de modification de configuration			M	M (=)
ExtStatus d'appareil			M	M (=)
Code de diffusion			M	M (=)
Profil d'appareil			M	M (=)
Code de réponse			M	M (=)
Etat de processus d'application			M	M (=)
Erreur de communication				S
Etat de communication				M
Etat de processus d'application				C
Aucune correspondance				S
Résultat (-)			S	S (=)
Code de réponse			M	M (=)
Etat de processus d'application			M	M (=)

NOTE La méthode de corrélation d'une primitive de confirmation avec sa primitive de demande précédente correspondante représente une question d'ordre local. La méthode de corrélation d'une primitive de réponse avec sa primitive d'indication précédente correspondante représente une question d'ordre local.

Argument

Il comporte les paramètres de la demande de service.

Balise

Ce paramètre est utilisé pour identifier l'appareil distant. Si la valeur d'AREP ID est l'adresse de diffusion, ce paramètre est obligatoire; il est facultatif dans le cas contraire.

Résultat (+)

Ce paramètre de type de sélection indique l'aboutissement positif (succès) de la demande de service. Les paramètres applicables à cette sélection sont les attributs de l'objet de classe VFD tel que décrit en 6.2.1.2.

Erreur de communication

Ce paramètre de type de sélection indique l'échec de la demande de service du fait d'une erreur de communication. Les paramètres applicables à cette sélection indiquent la raison de cet échec.

Aucune correspondance

Cette sélection est utilisée lorsque l'indication comporte le paramètre Balise et ne correspond pas à la balise présente dans l'appareil de réponse.

Résultat (–)

Ce paramètre de type de sélection indique l'échec de la demande de service. Les paramètres applicables à cette sélection indiquent la raison de cet échec.

6.2.1.3.3 Procédure de service

La procédure de service confirmé spécifiée en 3.5.3.3 s'applique à ce service. Si la primitive d'indication spécifie une balise et si l'appareil de réponse ne trouve pas la balise de correspondance, alors aucune réponse n'est transmise.

6.2.2 Variable ASE (ASE de variable)

6.2.2.1 Vue d'ensemble

L'ASE de variable propose des services de lecture ou d'écriture d'un objet variable dans l'appareil serveur. Un numéro identifie la variable. Cet identifiant numérique décrit le type de données et la structure de l'objet dans leur intégralité. Le type d'objet peut être l'un des types définis en 5.3.

6.2.2.2 Modèle de variable commun

6.2.2.2.1 Variable simple

6.2.2.2.1.1 Modèle formel

L'objet de variable simple représente une variable unique caractérisée par un type de données défini.

ASE:	VARIABLE ASE
CLASS:	VARIABLE simple
PARENT CLASS:	TOP
ATTRIBUTES:	
1 (m) Attribut clé:	Identifiant numérique
2 (o) Attribut:	Nom de variable
3 (m) Attribut:	Nom de type de données
SERVICES:	
1 (o) OpsService:	Lecture
2 (o) OpsService:	Ecriture
3 (o) OpsService:	Rapport d'information

6.2.2.2.1.2 Attributs

Identifiant numérique

Identifie une instance de cette classe d'objets.

Nom de variable

Nom attribué à l'objet.

Nom de type de données

Nom attribué au type de données spécifié en 5.3.1.

6.2.2.2.1.3 Services**Lecture**

Ce service permet à un client de lire la valeur d'une variable.

Ecriture

Ce service permet à un client d'écrire la valeur d'une variable.

Rapport d'information

Ce service non confirmé permet à un serveur d'envoyer la valeur d'une variable à un client quelconque ou à tous les clients.

6.2.2.2.2 Variable de structure**6.2.2.2.2.1 Modèle formel**

L'objet de structure consiste en un ensemble de variables simples de types de données différents: Un accès complet à l'objet d'enregistrement est possible, un accès séparé à ses champs individuels n'étant toutefois pas possible.

ASE:**VARIABLE ASE****CLASS:**

Structure **VARIABLE** (Variable de structure)

PARENT CLASS:

TOP

ATTRIBUTES:

- | | | | |
|-----|-----|---------------|---------------------------------|
| 1 | (m) | Attribut clé: | Identifiant numérique |
| 2 | (o) | Attribut: | Nom de variable |
| 3 | (m) | Attribut: | Nombre de champs |
| 4 | (m) | Attribut: | Liste de champs |
| 4.1 | (o) | Attribut: | Nom de champ |
| 4.2 | (m) | Attribut: | Nom de type de données de champ |

SERVICES:

- | | | | |
|---|-----|-------------|-----------------------|
| 1 | (o) | OpsService: | Lecture |
| 2 | (o) | OpsService: | Ecriture |
| 3 | (o) | OpsService: | Rapport d'information |

6.2.2.2.2.2 Attributs**Identifiant numérique**

Identifie une instance de cette classe d'objets.

Nom de variable

Nom attribué à l'objet.

Nombre de champs

Cet attribut définit le nombre de champs dans une structure.

Nom de champ

Cet attribut facultatif spécifie le nom du champ.

Nom de type de données de champ

Nom attribué au type de données spécifié en 5.3.1.

6.2.2.2.3 Services

Lecture

Ce service permet à un client de lire la valeur d'une variable.

Écriture

Ce service permet à un client d'écrire la valeur d'une variable.

Rapport d'information

Ce service non confirmé permet à un serveur d'envoyer la valeur d'une variable à un client quelconque ou à tous les clients.

6.2.2.2.3 Matrice de variable simple

6.2.2.2.3.1 Modèle formel

Cet objet de matrice est utilisé pour définir une variable construite dans laquelle tous les éléments ont le même type et la même longueur de données et sont de la classe de variable simple.

ASE:	VARIABLE ASE
CLASS:	Matrice de VARIABLE Simple
PARENT CLASS:	VARIABLE Simple
ATTRIBUTES:	
4 (m) Attribut:	Taille de la matrice

6.2.2.2.3.2 Attributs

Taille de la matrice

Il indique le nombre d'éléments contenus dans la matrice. Si la valeur de cet attribut est « Variable », le nombre d'éléments n'est pas fixé.

6.2.2.3 Spécification de service ASE variable

6.2.2.3.1 Services pris en charge

Ce paragraphe comprend la définition des services uniques pour cet ASE. Les services définis pour cet ASE sont les suivants:

- Lecture de la variable provenant de l'appareil serveur,
- Ecriture de la variable à destination de l'appareil serveur, et
- Transmission du rapport d'information provenant d'un appareil serveur.

Il existe des services distincts pour chaque variable simple ou d'enregistrement identifiée par un numéro. Le format des services est commun à toutes les variables. Le format commun du service Lecture est défini en 6.2.2.3.2. Le format commun du service Ecriture est défini en 6.2.2.3.4. Le format commun du service Rapport d'information est défini en 6.2.2.3.5. Le format de chaque variable est spécifié dans le service individuel de lecture ou d'écriture de cette variable.

NOTE Les définitions de service dans le présent paragraphe ne sont pas indépendantes de la variable concernée. Ce paragraphe présente uniquement le format du service. Les définitions propres au service comprennent le modèle de variable et les paramètres propres au service.

6.2.2.3.2 Corrélation des primitives

Au niveau du client, il n'y a qu'une seule demande en attente d'une valeur d'AREP ID. L'AREP ID est acheminé comme adresse de destination dans les primitives de service de la couche liaison de données. Par conséquent, la primitive de confirmation est corrélée avec la primitive de demande au moyen de l'AREP ID. Au niveau du serveur, il n'y a qu'une seule

primitive d'indication en attente à tout moment. Par conséquent, la primitive de réponse est corrélée avec la primitive d'indication en attente de la réponse.

6.2.2.3.3 Service Lecture – format commun

6.2.2.3.3.1 Vue d'ensemble du service

Ce service permet à un client de lire la valeur d'une variable émise par le serveur.

6.2.2.3.3.2 Primitives de service

Les paramètres de service applicables à ce service sont présentés dans le Tableau 7.

Tableau 7 – Paramètres de service Lecture

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Identifiant numérique	M	M (=)	M (=)	M (=)
Identifiant complémentaire	C	M (=)	M (=)	M (=)
Sous-indice	C	M (=)	M (=)	M (=)
Compte des préambules	M			
Résultat (+)			S	S (=)
Valeur de variable			M	M (=)
Code de réponse			M	M (=)
Etat de processus d'application			M	M (=)
Erreur de communication				S
Etat de communication				M
Etat de processus d'application				C
Résultat (-)			S	S (=)
Code de réponse			M	M (=)
Etat de processus d'application			M	M (=)

Argument

Il comporte les paramètres de la demande de service.

Identifiant numérique

Ce paramètre identifie la variable. Il s'agit de l'identifiant numérique attribué à la variable à lire.

Identifiant complémentaire

Ce paramètre fournit à certains des identifiants numériques l'information complémentaire requise pour identifier la/les variable(s) à lire.

Sous-indice

Ce paramètre identifie l'élément individuel dans une variable de matrice par sa position à l'intérieur de la variable. Il peut s'agir soit d'une valeur numérique soit d'une Enumération.

Valeur de variable

Ce paramètre spécifie la valeur de la variable lue à partir du serveur.

Etat de communication

Si cet état est «Appareil non disponible», la primitive de confirmation ne renvoie pas l'«Etat de processus d'application».

6.2.2.3.4 Service Ecriture – format commun

6.2.2.3.4.1 Vue d'ensemble du service

Ce service permet à un client d'écrire la valeur d'une variable sur le serveur.

6.2.2.3.4.2 Primitives de service

Les paramètres de service applicables à ce service sont présentés dans le Tableau 8.

Tableau 8 – Paramètres de service Ecriture

Nom du paramètre	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Identifiant numérique	M	M (=)	M	M (=)
Identifiant complémentaire	C	M (=)	M	M (=)
Sous-indice	C	M (=)	M	M (=)
Compte des préambules	M			
Valeur de variable	M	M (=)		
Résultat (+)			S	S (=)
Valeur de variable			M	M (=)
Code de réponse			M	M (=)
Etat du processus d'application			M	M (=)
Erreur de communication				S
Etat de communication				M
Etat du processus d'application				C
Résultat (-)			S	S (=)
Code de réponse			M	M (=)
Etat du processus d'application			M	M (=)

Argument

Il comporte les paramètres de la demande de service.

Identifiant numérique

Ce paramètre identifie la variable. Il s'agit de l'attribut clé attribué à la variable à écrire. On l'appelle également «numéro de commande».

Identifiant complémentaire

Ce paramètre fournit à certains des identifiants numériques l'information complémentaire requise pour identifier la/les variable(s) à écrire.

Sous-indice

Ce paramètre identifie l'élément individuel dans une variable de matrice par sa position à l'intérieur de la variable. Il peut s'agir soit d'une valeur numérique soit d'une Enumération.

Valeur de variable

Dans la primitive de demande, ce paramètre spécifie la valeur souhaitée de la variable à écrire. Dans la primitive de réponse, ce paramètre indique la valeur réelle de la variable qui a été écrite. Dans certains cas, le serveur de réponse écrit une valeur différente de celle contenue dans la primitive d'indication. Par exemple, il peut arrondir la valeur à la valeur admissible la plus proche ou à la valeur la plus proche que le serveur peut prendre en charge.

NOTE La valeur retournée de la variable peut être utilisée par l'utilisateur de la FAL pour valider l'opération d'écriture.

État de communication

Si cet état est «Appareil non disponible», la primitive de confirmation ne renvoie pas l'«Etat de processus d'application».

6.2.2.3.5 Service Rapport d'information – format commun

6.2.2.3.5.1 Vue d'ensemble du service

Ce service permet à un esclave d'envoyer la valeur d'une variable et à un maître de la recevoir sans utiliser une demande provenant du maître. Il sert à publier des données de variables de façon cyclique.

6.2.2.3.5.2 Primitives de service

Les paramètres de service applicables à ce service sont présentés dans le Tableau 9.

Tableau 9 – Paramètres de service Rapport d'information

Nom du paramètre	Req	Ind
Argument		
AREP ID	M	M (=)
Identifiant numérique	M	M (=)
Identifiant complémentaire	C	M (=)
Sous-indice	C	M (=)
Valeur de variable	M	M (=)
Code de réponse	M	M (=)
Etat du processus d'application	M	M (=)
Etat étendu	C	M(=)
Valeur de variable	M	M (=)

Argument

L'argument comporte les paramètres de la demande de service.

Identifiant numérique

Ce paramètre identifie la variable. Il s'agit de l'attribut clé attribué à la variable à lire. On l'appelle également «numéro de commande».

Identifiant complémentaire

Ce paramètre fournit à certains des identifiants numériques l'information complémentaire requise pour identifier la/les variable(s) à lire.

Sous-indice

Ce paramètre identifie l'élément individuel dans une variable de matrice par sa position à l'intérieur de la variable. Il peut s'agir soit d'une valeur numérique soit d'une Enumération.

Valeur de variable

Ce paramètre spécifie la valeur rapportée par le serveur.

6.2.3 Action ASE (ASE d'action)

6.2.3.1 Vue d'ensemble du service

Ce service permet de commander une action spécifiée du serveur. Le format commun du service Action est défini dans le présent article. Pour chaque action, Il existe un service distinct.

NOTE Les définitions de service dans le présent paragraphe ne sont pas indépendantes de l'action demandée. Ce paragraphe présente uniquement le format du service. Les définitions propres au service comprennent le comportement de l'action et les paramètres propres au service.

6.2.3.2 Corrélation des primitives

Au niveau du client, il n'y a qu'une seule demande en attente d'une valeur de l'AREP ID. L'AREP ID est acheminé comme adresse de destination dans les primitives de service de la couche transport. Par conséquent, la primitive de confirmation est corrélée avec la primitive de demande au moyen de l'AREP ID. Au niveau du serveur, il n'y a qu'une seule primitive d'indication en attente à tout moment. Par conséquent, la primitive de réponse est corrélée avec la primitive d'indication en attente de la réponse.

6.2.3.3 Modèle d'action

6.2.3.3.1 Généralités

Certaines actions nécessitent une variable pour valider l'action. Le modèle de ces actions est défini dans le présent paragraphe.

6.2.3.3.2 Modèle de format

ASE: Action simple
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Attribut clé: Identifiant numérique

ASE: Action qualifiée
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Attribut clé: Identifiant numérique
 2 (m) Attribut : Nom de qualificatif
 3 (m) Attribut Nom de type de données

ASE: Action de Variable simple
PARENT CLASS: TOP
ATTRIBUTES:
 1 (m) Attribut clé: Identifiant numérique
 2 (m) Attribut: Nom de variable
 3 (m) Attribut: Nom de type de données