

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 couche application – Éléments de Type 20**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2010 IEC, Geneva, Switzerland

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

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

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

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

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

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

About the IEC

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

About IEC publications

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

Useful links:

IEC publications search - www.iec.ch/searchpub

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

It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Customer Service Centre - webstore.iec.ch/csc

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

A propos de la CEI

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

A propos des publications CEI

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

Liens utiles:

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

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

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

Just Published CEI - webstore.iec.ch/justpublished

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

Electropedia - www.electropedia.org

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

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE

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

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX



ICS 25.04.40; 35.100.70; 35.110

ISBN 978-2-88912-864-8

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

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
1.1 Overview.....	7
1.2 Specifications.....	8
1.3 Conformance.....	8
2 Normative references.....	8
3 Terms, definitions, symbols, abbreviations and conventions.....	9
3.1 ISO/IEC 7498-1 terms.....	9
3.2 ISO/IEC 8822 terms.....	9
3.3 ISO/IEC 9545 terms.....	9
3.4 ISO/IEC 8824 terms.....	9
3.5 IEC/TR 61158-1 terms.....	9
3.6 Fieldbus application-layer specific definitions.....	12
3.7 Abbreviations and symbols.....	14
3.8 Conventions.....	15
4 Concepts.....	18
5 Data type ASE.....	18
5.1 Overview.....	18
5.2 Formal definition of data type objects.....	20
5.3 FAL defined data types.....	21
5.4 Data type ASE service specification.....	25
5.5 Summary of data types.....	25
6 Communication model specification.....	25
6.1 Common parameters.....	25
6.2 ASEs.....	27
6.3 ARs.....	43
6.4 Summary of classes.....	46
6.5 Permitted services by AREP role.....	46
Bibliography.....	47
Figure 1 – Data type class hierarchy.....	19
Figure 2 – VFD model.....	27
Table 1 – Packed ASCII character set.....	24
Table 2 – ISO Latin-1 characters.....	25
Table 3 – Data type summary.....	25
Table 4 – Comm status values.....	26
Table 5 – Response code values.....	26
Table 6 – Device status values.....	26
Table 7 – Identify service parameters.....	30
Table 8 – Read service parameters.....	33
Table 9 – Write service parameters.....	34
Table 10 – AR get attributes service parameters.....	44

Table 11 – AR set attributes service parameters	45
Table 12 – Class summary.....	46
Table 13 – Services by AREP class	46

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2010

Withdrawn

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.

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

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

The main change with respect to the previous edition is listed below:

- Revised VFD ASE, see 6.2.1.

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

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

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

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

The French version has not been voted upon.

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

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

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

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

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

INTRODUCTION

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

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

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

Withdrawing
IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2010

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-20: Application layer service definition – Type 20 elements

1 Scope

1.1 Overview

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

This part of IEC 61158 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 standard defines in an abstract way the externally visible service provided by the different Types of the fieldbus Application Layer in terms of

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

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

- a) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and
- b) Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

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

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

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

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

1.2 Specifications

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

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

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

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

1.3 Conformance

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

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfill the Type 20 application layer services as defined in this standard.

2 Normative references

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

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

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

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

ISO/IEC 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*

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 ISO/IEC 7498-1 terms

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

3.2 ISO/IEC 8822 terms

- a) abstract syntax
- b) presentation context

3.3 ISO/IEC 9545 terms

- a) application-entity-invocation
- b) application-entity-type
- c) application-process-invocation
- d) application-process-type
- e) application-service-elementt

3.4 ISO/IEC 8824 terms

- a) object identifier
- b) type

3.5 IEC/TR 61158-1 terms

The following IEC/TR 61158-1 terms apply.

3.5.1

application

function of data structure for which data is consumed or produced

3.5.2

application layer interoperability

capability of application entities to perform coordinated and cooperative operations using the services of the FAL

3.5.3

application object

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

NOTE Multiple types of application object classes may be defined.

3.5.4

application process

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

3.5.5

application process identifier

distinguishes multiple application processes used in a device

3.5.6

application process object

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

3.5.7

application process object class

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

3.5.8

application relationship

cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of pre-configuration activities

3.5.9

application relationship application service element

application-service-element that provides the exclusive means for establishing and terminating all application relationships

3.5.10

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. Each application process involved in the application relationship maintains its own application relationship endpoint

3.5.11

attribute

description of an externally visible characteristic or feature of an object. 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

3.5.12

behaviour

indication of how the object responds to particular events. Its description includes the relationship between attribute values and services

3.5.13**class**

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

3.5.14**class attributes**

an attribute that is shared by all objects within the same class

3.5.15**class code**

a unique identifier assigned to each object class

3.5.16**class specific service**

a service defined by a particular object class to perform a required function which is not performed by a common service. A class specific object is unique to the object class which defines it

3.5.17**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

3.5.18**conveyance path**

unidirectional flow of APDUs across an application relationship

3.5.19**cyclic**

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

3.5.20**dedicated AR**

AR used directly by the FAL User. On Dedicated ARs, only the FAL Header and the user data are transferred

3.5.21**device**

a physical hardware connection to the link. A device may contain more than one node

3.5.22**device profile**

a collection of device dependent information and functionality providing consistency between similar devices of the same device type

3.5.23**endpoint**

one of the communicating entities involved in a connection

3.5.24**error**

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

3.5.25

error code

identification of a specific type of error within an error class

3.5.26

management information

network-accessible information that supports managing the operation of the fieldbus system, including the application layer. Managing includes functions such as controlling, monitoring, and diagnosing

3.5.27

network

a series of nodes connected by some type of communication medium. The connection paths between any pair of nodes can include repeaters, routers and gateways

3.5.28

pre-defined AR endpoint

AR endpoint that is defined locally within a device without use of the create service. Pre-defined ARs that are not pre-established are established before being used

3.5.29

pre-established AR endpoint

AR endpoint that is placed in an established state during configuration of the AEs that control its endpoints

3.5.30

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

3.5.31

service

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

3.6 Fieldbus application-layer specific definitions

There are additional terms defined for this part.

3.6.1 analog channel

A continuously varying electrical signal connecting a field device to the remainder of the data acquisition or control system. Some field devices support multiple analog channels (input or output). Each analog channel transmits a single dynamic variable to or from the field device.

3.6.2 broadcast address

A broadcast address is used by a master to send a command to all devices. The broadcast address is five octet long and has all zeros as the value.

3.6.3 busy

The device is busy and cannot execute a command at the time. A device indicates busy by returning response code 32 when allowed by the command specification. The requested command is not executed if a busy response is returned.

3.6.4 device ID

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

3.6.5 device type

This 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.

3.6.6 device variable

A uniquely defined data item within a Field Device that is always associated with cyclical process information. A device variable's value varies in response to changes and variations in the process to which the device is connected.

3.6.7 dynamic variable

The connection between the process and an analog channel. A device may contain primary, secondary, tertiary, and quaternary variables. These are collectively called the dynamic variables.

3.6.8 long tag

It is a 32 character ISO Latin-1 string used to identify a field device.

3.6.9 loop current

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

3.6.10 manufacturer ID

This 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.

3.6.11 master

A device that initiates communication activity by sending request PDU to a device.

3.6.12 polling address

An integer used to identify the device. The polling address is used to construct one octet long address.

3.6.13 slave

A 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.

3.6.14 tag

The tag is a 8 character ASCII string used to identify the field device.

3.6.15 unique address

This is a five octet long address of a device and it uniquely identifies the device among all other devices that support this standard. It is a concatenation of the manufacturer ID, device type and device ID.

3.7 Abbreviations and symbols

AE	Application Entity
AL	Application Layer
ALP	Application Layer Protocol
APO	Application Object
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
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

SAP	Service Access Point
SDU	Service Data Unit
VFD	Virtual Field Device

3.8 Conventions

3.8.1 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.

3.8.2 Conventions for class definitions

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

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

- (1) The "FAL ASE:" entry is the name of the FAL ASE that provides the services for the class being specified.
- (2) The "CLASS:" entry is the name of the class being specified. All objects defined using this template will be an instance of this class. The class may be specified by this standard, or by a user of this standard.
- (3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They

have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2048 are allocated for identifying user defined classes.

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

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

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

- a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.
- b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.
- c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify
 - i) fields of a structured attribute (4.1, 4.2, 4.3),
 - ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).
 - iii) the selection fields of a choice type attribute (6.1 and 6.2).

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

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

3.8.3 Conventions for service definitions

3.8.3.1 General

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

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

3.8.3.2 Service parameters

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

NOTE 1 See the note under 3.8.3.3 relative to the non-inclusion of service parameters that are appropriate to a protocol specification or programming interface specification or implementation specification, but not to an abstract service definition.

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

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

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

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

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

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.

3.8.3.3 Service procedures

The procedures are defined in terms of

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

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

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

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

Further information on the conceptual service procedures of an implementation of a protocol that realizes the services of one of the IEC 61158-5 subseries abstract service definitions can be found in IEC/TR 61158-1:2010, 9.6.

4 Concepts

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

5 Data type ASE

5.1 Overview

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 standard.

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

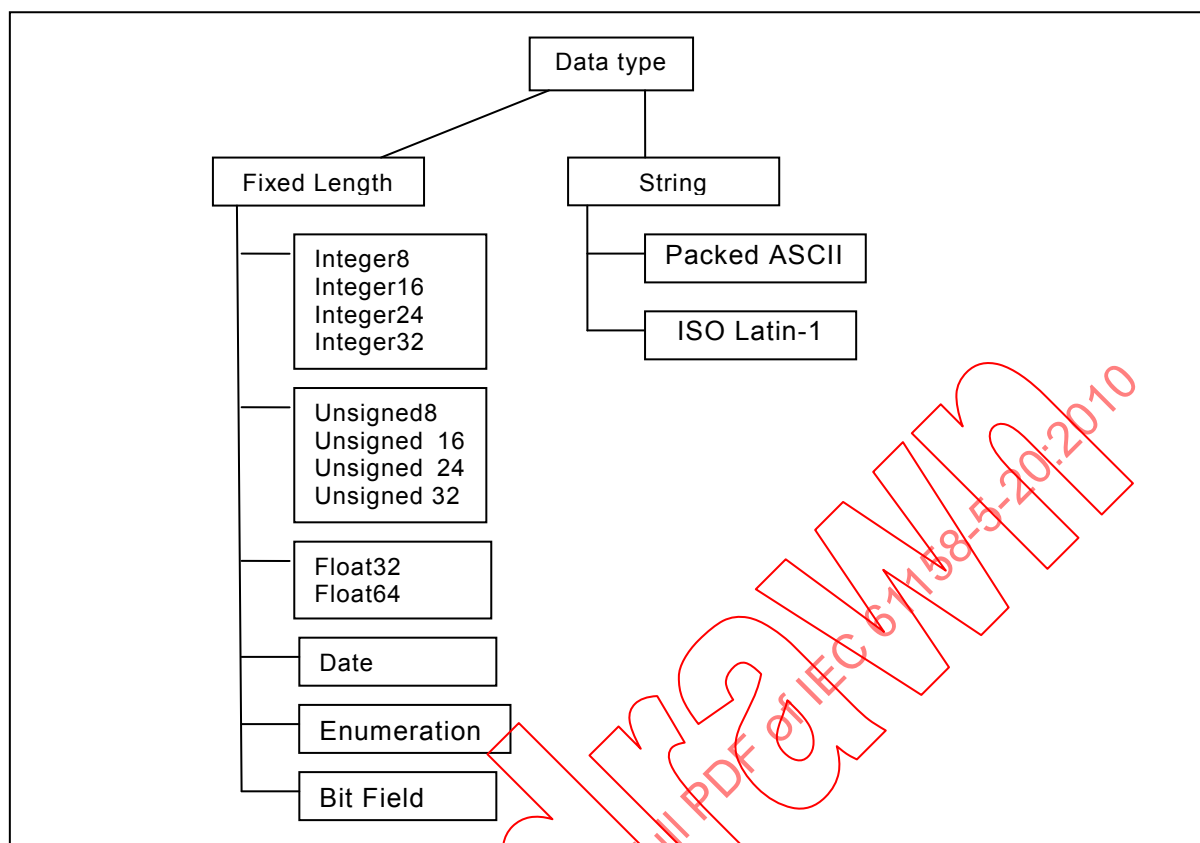


Figure 1 – Data type class hierarchy

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 used to define fixed length data types. Standard types taken from ISO/IEC 8824 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 standard are strings, arrays and structures. There are no standard types defined for arrays and structures.

5.1.1 Overview Of basic types

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

Simple types are ISO/IEC 8824 universal types. They are defined in this standard 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 standard for the purpose of assigning Fieldbus class identifiers to them. This standard does not change their definitions as specified in ISO/IEC 8824.

5.1.1.1 Fixed length types

The length of Fixed length types is an integral number of octets.

5.1.2 Overview of constructed types

Constructed data types are needed to completely convey the variety of information present on the Fieldbus. There are two kinds of constructed types defined for this standard, arrays and structures.

5.1.2.1 Strings

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

5.1.3 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 standard.

5.1.4 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	(o) Key Attribute:	Data type Numeric Identifier
2	(m) Key Attribute:	Data type Name
3	(m) Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c) Constraint:	Format = FIXED LENGTH STRING
4.1	(m) Attribute:	Octet Length
5	(c) Constraint:	Format = STRUCTURE
5.1	(m) Attribute:	Number of Fields
5.2	(m) Attribute:	List of Fields
5.2.1	(o) Attribute:	Field Name
5.2.2	(m) Attribute:	Field Data type
6	(c) Constraint:	Format = ARRAY
6.1	(m) Attribute:	Number of Array Elements
6.2	(m) Attribute :	Array Element Data type

5.2.1.1 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 supported by identifying the field by number. This attribute is present when the value of the format attribute is "STRUCTURE".

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 either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the Embedded Data type description shown below is used.

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 either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here.

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Integer8

CLASS:		Data type
ATTRIBUTES:		
1	Data type Numeric Identifier	= Not used
2	Data type Name	= Integer8
3	Format	= FIXED LENGTH
4.1	Octet Length	= 1

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

5.3.1.2 Integer16

CLASS:		Data type
ATTRIBUTES:		

1	Data type Numeric Identifier	=	Not used
2	Data type Name	=	Integer16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

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

5.3.1.3 Integer24

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	Not used
2	Data type Name	=	Integer24
3	Format	=	FIXED LENGTH
4.1	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 Numeric Identifier	=	Not used
2	Data type Name	=	Integer32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

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

5.3.1.5 Unsigned8

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	Not used
2	Data type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

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

5.3.1.6 Unsigned16

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	Not used
2	Data type Name	=	Unsigned16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

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

5.3.1.7 Unsigned24

CLASS: Data type

ATTRIBUTES:

1	Data type Numeric Identifier	=	Not used
2	Data type Name	=	Unsigned24
3	Format	=	FIXED LENGTH

4.1 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 Numeric Identifier = Not used
 2 Data type Name = Unsigned32
 3 Format = FIXED LENGTH
 4.1 Octet Length = 4

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

5.3.1.9 Float32

CLASS: Data type

ATTRIBUTES:

1 Data type Numeric Identifier = Not used
 2 Data type Name = Float32
 4 Format = FIXED LENGTH
 4.1 Octet Length = 4

This type has a length of four octets. The format for Float32 is that defined by ANSI/IEEE 754 as single precision.

5.3.1.10 Float64

CLASS: Data type

ATTRIBUTES:

1 Data type Numeric Identifier = Not used
 2 Data type Name = Float64
 4 Format = FIXED LENGTH
 4.1 Octet Length = 8

This type has a length of eight octets. The format for Float64 is that defined by ANSI/IEEE 754 as double precision.

5.3.1.11 Date

CLASS: Data type

ATTRIBUTES:

1 Data type Numeric Identifier = Not used
 2 Data type Name = Date
 4 Format = FIXED LENGTH
 4.1 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 Numeric Identifier = Not used
 2 Data type Name = Enumeration
 3 Format = FIXED LENGTH
 4.1 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 Numeric Identifier = Not used
- 2 Data type Name = Bit Field
- 3 Format = FIXED LENGTH
- 5.1 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.

5.3.2 String Types

5.3.2.1 Packed ASCII

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = Not used
- 2 Data type Name = Packed ASCII
- 3 Format = STRING
- 4.1 Octet Length = 1 to n

This type is a modified subset of the ASCII character code set. This subset is shown in the 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	:	;	<	=	>	?

^a SP indicates a space character.

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

5.3.2.2 ISO Latin-1

CLASS: Data type

ATTRIBUTES:

- 1 Data type Numeric Identifier = Not used
- 2 Data type Name = Latin-1
- 3 Format = STRING
- 4.1 Octet Length = 1 to n

This type is a subset of the ISO 8859-1 (ISO Latin-1) string type. The subset is shown in the 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	ç	£	¤	¥	¦	§	¨	©	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

^a SP indicates a space character.

^b NBSP indicates a non-breaking space character.

^c SHY indicates a soft hyphen.

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.

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
Integer8	5.3.1.1
Integer16	5.3.1.2
Integer24	5.3.1.3
Integer32	5.3.1.4
Unsigned8	5.3.1.5
Unsigned16	5.3.1.6
Unsigned24	5.3.1.7
Unsigned32	5.3.1.8

Data type	Clause
Float32	5.3.1.9
Float64	5.3.1.10
Date	5.3.1.11
Enumeration	5.3.1.12
Bit Field	5.3.1.13
Packed ASCII	5.3.2.1
ISO Latin-1	5.3.2.2

6 Communication model specification

6.1 Common parameters

Many services have the following parameters. Instead of defining them with each service, the following common definitions are provided.

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.

Comm Status

This parameter specifies information about the communication failure. The possible values of this parameter are shown in the Table 4.

Table 4 – Comm 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 UART was overwritten before it was read (i.e., the slave 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 UART (i.e. a mark or 1 was not detected when a Stop bit should have occurred).
Longitudinal Parity Error	The Longitudinal Parity calculated by the device did not match the Check Octet at the end of the message.
Buffer Overflow	The message was too long for the receive buffer of the device.
Device not available	The client did not receive any response from the server.

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 the Table 5.

Table 5 – 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).

Device Status

This parameter indicates the current operating status of the device as a whole and it is not associated with the completion of any command. The possible values of this parameter are shown in the Table 6.

Table 6 – Device status values

Value	Description
Device Malfunction	The device detected a serious error or failure that compromises device operation.
Configuration Changed	An operation was performed that changed the device's configuration.
Cold Start	A power failure or Device Reset has occurred.
More Status Available	More status information is available via another Read Command.
Loop Current Fixed	The Loop Current is being held at a fixed value and is not responding to process variations.
Loop Current Saturated	The Loop Current has reached its upper (or lower) endpoint limit and cannot increase (or decrease) any further.
Non-Primary Variable Out of Limits	A Device Variable not mapped to the Primary Variable is beyond its operating limits.
Primary Variable Out of Limits	The Primary Variable is beyond its operating limit.

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.

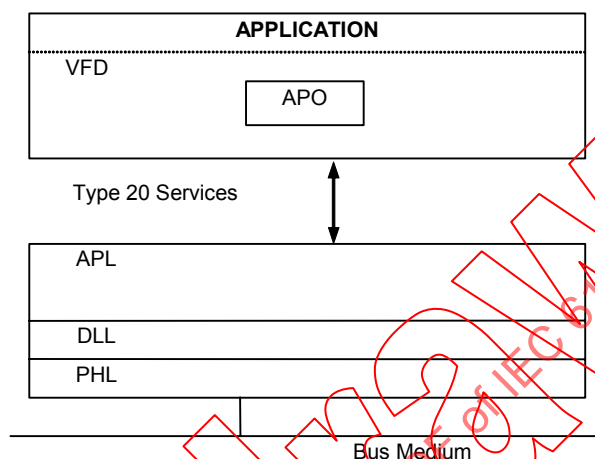


Figure 2 – VFD model

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 standard. 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 7.

Table 7 – 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 (=)
Device Status			M	M (=)
Comm Error				S
Comm Status				M
Device Status				M
No Match				S
Result (–)			S	S (=)
Response Code			M	M (=)
Device 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 paramter 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.

Comm 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 3.8.3.3 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

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
4 (o) Attribute:	Length
SERVICES:	
1 (o) OpService:	Read
2 (o) OpService:	Write

6.2.2.2.1.1 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.

Length

Length in octets of the data.

6.2.2.2.1.2 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.

6.2.2.2.2 Record variable

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

ASE:	VARIABLE ASE
CLASS:	RECORD 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

6.2.2.2.2.1 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.

6.2.2.2.2.2 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.

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.

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 6.2.2.3.3. The format of each variable is specified in the individual service to read or write that variable.

6.2.2.3.2 Read service – common format

6.2.2.3.2.1 Service overview

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

6.2.2.3.2.2 Service primitives

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

Table 8 – Read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP ID	M	M (=)		
Variable Index	M	M (=)		
Variable Sub Index	C	M (=)		
Result (+)			S	S (=)
Data			S	S (=)
Response Code			M	M (=)
Device Status			M	M (=)
Comm Error				S
Comm Status				M
Device Status				M
Result (–)			S	S (=)
Response Code			M	M (=)
Device 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.

Variable Index

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

Variable Sub Index

This parameter combined with Variable Index identifies the variable. This is the Variable Code assigned to the variable to be read. This parameter is present only for some specific values of Variable Index.

Result (+)

Data

This parameter specifies the value of the variable.

6.2.2.3.3 Write service – common format

6.2.2.3.3.1.1 Service overview

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

6.2.2.3.3.1.2 Service primitives

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

Table 9 – Write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M (=)		
AREP ID	M	M (=)		
Variable Index	M	M (=)		
Data	M	M (=)		
Result (+)			S	S (=)
Data			S	S (=)
Response Code			M	M (=)
Device Status			M	M (=)
Comm Error				S
Comm Status				M
Device Status				M
Result (–)			S	S (=)
Response Code			M	M (=)
Device 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.

Variable Index

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

Data

This parameter specifies the value of the variable.

6.2.2.3.4 Read primary variable

6.2.2.3.4.1 Service overview

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

6.2.2.3.4.2 Variable model

CLASS:	RECORD 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	= Enumeration
5.3 Field Name	= Primary variable
5.4 Field Data type Name	= Float32

6.2.2.3.5 Read loop current and percent of range

6.2.2.3.5.1 Service overview

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

6.2.2.3.5.2 Variable model

CLASS:	RECORD 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

6.2.2.3.6 Read dynamic variables and loop current

6.2.2.3.6.1 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.

6.2.2.3.6.2 Variable model

CLASS:	RECORD 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 = Enumeration
 - 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 = Enumeration
 - 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 = Enumeration
 - 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 = Enumeration
 - 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 3.1.1 to 3.1.6 are always present. If the number of fields is 5 or 7 or 9, then fields 3.2.1 to 3.2.4 are present. If the number of fields is 7 or 9, then fields 3.3.1 to 3.3.4 are present. If the number of fields is 9, then fields 3.4.1 to 3.4.4 are present.

6.2.2.3.7 Write loop configuration

6.2.2.3.7.1 Service overview

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

6.2.2.3.7.2 Variable model

CLASS: RECORD 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

6.2.2.3.8 Read loop configuration

6.2.2.3.8.1 Service overview

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

6.2.2.3.8.2 Variable model

CLASS: **RECORD 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

6.2.2.3.9 Read dynamic variable families classifications

6.2.2.3.9.1 Service overview

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

6.2.2.3.9.2 Variable model

CLASS: **RECORD 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

6.2.2.3.10 Read device variables with status

6.2.2.3.10.1 Service overview

This service permits a client to read the value of up to four Device or Dynamic variable from the server. Client may request only 1, 2, 3 or 4 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.

6.2.2.3.10.2 Request Variable Sub Index model

CLASS: RECORD VARIABLE

ATTRIBUTES:

- 1 (c) Constraint Variable Sub Index = up to 4
- 1.1 (m) Sub Index 1
- 1.1.1 (m) Field Name = Variable code
- 1.1.2 (m) Field Data type Name = Unsigned8
- 1.2 (c) Sub Index 2
- 1.2.1 (m) Field Name = Variable code
- 1.2.2 (m) Field Data type Name = Unsigned8
- 1.3 (c) Sub Index 3
- 1.3.1 (m) Field Name = Variable code
- 1.3.2 (m) Field Data type Name = Unsigned8
- 1.4 (c) Sub Index 4
- 1.4.1 (m) Field Name = Variable code
- 1.4.2 (m) Field Data type Name = Unsigned8

6.2.2.3.10.3 Response Variable model

CLASS: RECORD 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}
- 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 = Enumeration
- 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}
- 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 = Enumeration
- 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}
- 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	= Enumeration
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
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	= Enumeration
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

The number of variables can be either 1 or 2 or 3 or 4. The fields marked as 3.1.1 to 3.1.12 are always present. If the number of fields is 2 or 3 or 4, then fields 3.2.1 to 3.2.10 are present. If the number of fields is 3 or 4, then fields 3.3.1 to 3.3.10 are present. If the number of fields is 4, then fields 3.4.1 to 3.4.10 are present.

6.2.2.3.11 Read message

6.2.2.3.11.1 Service overview

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

6.2.2.3.11.2 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

6.2.2.3.12 Read tag, descriptor and date

6.2.2.3.12.1 Service overview

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

6.2.2.3.12.2 Variable model

CLASS: RECORD 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

6.2.2.3.13 Read primary variable transducer information

6.2.2.3.13.1 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.

6.2.2.3.13.2 Variable model

CLASS: RECORD 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 = Enumeration
 - 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

6.2.2.3.14 Read device information

6.2.2.3.14.1 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.

6.2.2.3.14.2 Variable model

CLASS: RECORD 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 upper and lower range values Unit

5.6	Field Data type Name	= Enumeration
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	= Private label distributor
5.16	Field Data type Name	= Enumeration
5.17	Field Name	= PV analog channel
5.18	Field Data type Name	= Bit Field

6.2.2.3.15 Read final assembly number

6.2.2.3.15.1 Service overview

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

6.2.2.3.15.2 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

6.2.2.3.16 Write message

6.2.2.3.16.1 Service overview

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

6.2.2.3.16.2 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

6.2.2.3.17 Write tag, descriptor and date

6.2.2.3.17.1 Service overview

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

6.2.2.3.17.2 Variable model

CLASS: RECORD 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

6.2.2.3.18 Write final assembly number

6.2.2.3.18.1 Service overview

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

6.2.2.3.18.2 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

6.2.2.3.19 Read long tag

6.2.2.3.19.1 Service overview

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

6.2.2.3.19.2 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

6.2.2.3.20 Write long tag

6.2.2.3.20.1 Service overview

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

6.2.2.3.20.2 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 |

6.3 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 Mster 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) OpService:	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 PDU 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 10.

Table 10 – 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. See 1.2.		

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 11.

Table 11 – 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. See 1.2.		

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 12 provides a summary of the classes.

Table 12 – Class summary

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

6.5 Permitted services by AREP role

Table 13 defines the valid combinations of services and AREP class (which service APDUs and AREP with the specified class can send or receive). All services are confirmed.

Table 13 – Services by AREP class

Services	Master		Slave	
	req	rcv	req	rcv
VFD ASE				
Identify		X		X
Variable ASE				
Read service – common	X			X
Write service – common	X			X
Read Primary variable	X			X
Read Loop current And Percent Of range	X			X
Read Dynamic variables and Loop current	X			X
Write Polling address	X			X
Read Loop configuration	X			X
Read Dynamic variable families classifications	X			X
Read Device variables with status	X			X
Read Message	X			X
Read Tag, Descriptor, Date	X			X
Read Primary variable transducer information	X			X
Read Device information	X			X
Read Final assembly number	X			X
Write Message	X			X
Write Tag, Descriptor, Date	X			X
Write Final assembly number	X			X
Read Long Tag	X			X
Write Long Tag	X			X

Bibliography

IEC/TR 61158-1:2010¹, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

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

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

¹ To be published

SOMMAIRE

AVANT-PROPOS	50
INTRODUCTION.....	52
1 Domaine d'application	53
1.1 Vue d'ensemble.....	53
1.2 Spécifications	54
1.3 Conformité	54
2 Références normatives	54
3 Termes, définitions, symboles, abréviations et conventions	55
3.1 Termes de l'ISO/CEI 7498-1	55
3.2 Termes de l'ISO/CEI 8822	55
3.3 Termes de l'ISO/CEI 9545	55
3.4 Termes de l'ISO/CEI 8824	55
3.5 Termes de la IEC/TR 61158-1	55
3.6 Définitions spécifiques à la couche application de bus de terrain.....	58
3.7 Abréviations et symboles.....	60
3.8 Conventions	61
4 Concepts	64
5 ASE des types de données	64
5.1 Vue d'ensemble.....	64
5.2 Définition formelle des objets de types de données	66
5.3 Types de données définis par la FAL.....	68
5.4 Spécification des services ASE pour le type de données	71
5.5 Synthèse des types de données	72
6 Spécification du modèle de communication	72
6.1 Paramètres communs	72
6.2 ASE.....	73
6.3 AR.....	89
6.4 Synthèse des classes.....	92
6.5 Services admis par le rôle de l'AREP	92
Bibliographie.....	94
Figure 1 – Hiérarchie des classes de types de données	65
Figure 2 – Modèle VFD	73
Tableau 1 – Ensemble de caractères ASCII compact	71
Tableau 2 – Caractères ISO Latin n°1	71
Tableau 3 – Synthèse des types de données	72
Tableau 4 – Valeurs d'état de communication	72
Tableau 5 – Valeurs de code de réponse	73
Tableau 6 – Valeurs d'état du dispositif.....	73
Tableau 7 – Paramètres de service d'identification	76
Tableau 8 – Paramètres de service Lecture	79
Tableau 9 – Paramètres de service Ecriture.....	80
Tableau 10 – Paramètres de service AR get attributes.....	91

Tableau 11 – Paramètres de service AR set attributes	92
Tableau 12 – Synthèse des classes	92
Tableau 13 – Services par classe d'AREP	93

IECNORM.COM : Click to view the full PDF of IEC 61158-5-20:2010

Withdrawn

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

Partie 5-20: Définition des services de 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 propriété intellectuelle ou de droits analogues. La CEI ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de propriété et de ne pas avoir signalé leur existence.

NOTE 1 L'utilisation de certains des types de protocoles est restreinte par les détenteurs des droits de propriété intellectuelle correspondants. Quoi qu'il en soit, l'engagement pris par les détenteurs, quant à une diffusion limitée desdits droits de propriété intellectuelle, permet d'utiliser un type particulier de protocole de Couche Liaison de données avec des protocoles de Couche Physique et de Couche Application dans les combinaisons de types explicitement spécifiées dans les parties de profils. L'utilisation des divers types de protocoles dans d'autres combinaisons peut nécessiter l'autorisation de leurs détenteurs de droits de propriété intellectuelle respectifs.

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

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

La principale modification par rapport à l'édition précédente est mentionnée ci-dessous:

- VFD ASE révisée, voir 6.2.1.

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

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

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

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

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

Une liste de toutes les parties de la série CEI 61158, présentées sous le titre général *Réseaux de communication industriels – Spécifications des bus de terrain*, est disponible 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.

NOTE 2 La révision de la présente norme fera l'objet d'une synchronisation avec les autres parties de la série CEI 61158.

INTRODUCTION

La présente norme fait partie d'une série élaborée pour faciliter l'interconnexion des composants de systèmes d'automatisation. Elle est apparentée à d'autres normes dans l'ensemble, comme défini par le modèle de référence de bus de terrain "à trois couches" décrit dans la IEC/TR 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. La présente norme 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 la présente norme 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 61158-5-2010

RÉSEAUX DE COMMUNICATION INDUSTRIELS – SPÉCIFICATIONS DES BUS DE TERRAIN –

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

1 Domaine d'application

1.1 Vue d'ensemble

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 partie de la CEI 61158 fournit des éléments communs pour les communications de messagerie de base, critiques et non critiques du point de vue temporel, entre les programmes d'application dans un environnement automatisé et les équipements spécifiques aux bus de terrain de Type 20. Le terme "critique du point de vue temporel" est utilisé pour représenter la présence d'une fenêtre temporelle, dans le cadre de laquelle une ou plusieurs actions spécifiées doivent 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 définit de manière abstraite le service visible externe fourni par les différents types 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 a pour objet de définir les services fournis

- a) à l'utilisateur FAL à la limite entre l'utilisateur et la couche application du modèle de référence de bus de terrain, et
- b) à la gestion des systèmes à la limite entre la couche application et la gestion des systèmes du modèle de référence de bus de terrain.

La présente norme 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) et à la structure de la couche application OSI (ISO/CEI 9545).

Les services et protocoles FAL sont fournis par les entités d'application (AE) FAL des processus d'application. L'AE de la FAL comprend un ensemble d'éléments de service d'application orientés objet (ASE) et une Entité de Gestion de Couches (LME) qui gère les AE. Les ASE fournissent des services de communication qui fonctionnent sur un ensemble de classes d'objets de processus d'application (APO) associées. Une des ASE de la FAL est une ASE de gestion qui fournit un ensemble commun de services dédiés à la gestion des instances des classes FAL.

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 définit également certains services d'appui destinés à fournir un accès à la FAL afin de contrôler certains aspects de son fonctionnement.

1.2 Spécifications

Le principal objectif de la présente norme est de spécifier les caractéristiques des services de couche application conceptuelle adaptés aux communications critiques du point de vue temporel, et de compléter ainsi le modèle de référence de base OSI dans l'orientation du développement des protocoles de couche application pour les communications critiques du point de vue temporel.

Le second objectif est de fournir des voies de migration issues des protocoles de communication industriels précédents. Ce dernier objectif génère la diversité des services normalisés comme les divers types de la CEI 61158.

Cette spécification peut servir de base pour les interfaces de programmation d'application formelles. Néanmoins, elle ne constitue pas une interface de programmation formelle, et toute interface de ce type devra traiter les questions de mise en œuvre non couvertes par cette spécification, y compris

- a) les dimensions et l'ordre des octets des divers paramètres de services à octets multiples, et
- b) la corrélation des primitives appariées de demande et de confirmation, ou d'indication et de réponse.

1.3 Conformité

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

Il n'y a pas de conformité des équipements à la présente norme de définition du service de couche application. En revanche, la conformité est obtenue par la mise en œuvre des protocoles de couche application conformes qui exécutent les services de couche application de Type 20, tel que défini dans la présente norme.

2 Références normatives

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

ISO/CEI 74981-1, *Technologies de l'information – Interconnexion de systèmes ouverts (OSI) – Modèle de référence de base: Le modèle de base*

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

ISO/CEI 8824-1, *Technologies de l'information – Notation de syntaxe abstraite numéro un (ASN.1): Spécification de la notation de base*

ISO/CEI 8859-1, *Technologies de l'information – Jeux de caractères graphiques codés sur un seul octet – Partie 1: Alphabet latin n° 1*

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

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

3 Termes, définitions, symboles, abréviations et conventions

Pour les besoins du présent document, les termes suivants tels que définis dans ces publications s'appliquent:

3.1 Termes de l'ISO/CEI 7498-1

- a) entité d'application
- b) processus d'application
- c) unité de données de protocole d'application
- d) élément de service d'application
- e) invocation de l'entité d'application
- f) invocation de processus d'application
- g) transaction d'application
- h) système ouvert réel
- i) syntaxe de transfert

3.2 Termes de l'ISO/CEI 8822

- a) syntaxe abstraite
- b) contexte de présentation

3.3 Termes de l'ISO/CEI 9545

- a) invocation de l'entité d'application
- b) type d'entité d'application
- c) invocation du processus d'application
- d) type de processus d'application
- e) élément de service d'application

3.4 Termes de l'ISO/CEI 8824

- a) identifiant d'objet
- b) type

3.5 Termes de la IEC/TR 61158-1

Les termes suivants de la IEC/TR 61158-1 s'appliquent.

3.5.1

application

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

3.5.2

interopérabilité de la couche application

capacité des entités d'application à effectuer des opérations coordonnées et coopératives en utilisant les services de la FAL

3.5.3

objet d'application

classe d'objet qui gère et permet l'échange de messages du temps de traitement sur le réseau et au sein du dispositif de réseau

NOTE Plusieurs types de classes d'objet d'application peuvent être définis.

3.5.4

processus d'application

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

3.5.5

identifiant du processus d'application

distingue les multiples processus d'application utilisés dans un dispositif

3.5.6

objet du processus d'application

composant d'un processus d'application identifiable et accessible par une relation d'application FAL. 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'objet 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

3.5.7

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

3.5.8

relation d'application

association de coopération entre deux invocations ou plus d'entités d'application en vue de l'échange d'informations et de la coordination de leur fonctionnement commun. Cette relation est activée soit par l'échange d'unités de données de protocole d'application ou suite à des activités de pré-configuration

3.5.9

élément de service d'application pour la relation d'application

élément de service d'application qui offre le moyen exclusif d'établir et d'achever toutes les relations d'application

3.5.10

extrémité de la relation d'application

contexte et comportement d'une relation d'application telle qu'observée et maintenue par l'un des processus d'application impliqués dans la relation d'application. Chaque processus d'application impliqué dans la relation d'application maintient sa propre extrémité de la relation d'application

3.5.11**attribut**

description d'une caractéristique ou fonction extérieurement visible d'un objet. Les attributs d'un objet contiennent les informations concernant les parties variables d'un objet. Généralement, ces attributs fournissent des informations d'état ou régissent le fonctionnement d'un objet. Les attributs peuvent également altérer le comportement d'un objet. Ces attributs sont répartis en attributs de classe et d'instance

3.5.12**comportement**

indication de la manière de répondre de l'objet à des événements particuliers. Sa description inclut la relation entre les valeurs et services d'attributs

3.5.13**classe**

ensemble d'objets, dont tous représentent le même type de composant système. Une classe est une généralisation de l'objet, et un modèle de définition des variables et des méthodes. Tous les objets d'une classe sont identiques en termes de forme et de comportement, mais leurs attributs contiennent habituellement des données différentes

3.5.14**attributs de classe**

attribut partagé par tous les objets de la même classe

3.5.15**code de classe**

identifiant unique affecté à chaque classe d'objets

3.5.16**service spécifique à la classe**

service défini par une classe d'objets particulière afin d'exécuter une fonction requise non exécutée par un service commun. Un objet spécifique à la classe est unique à la classe d'objets qui le définit

3.5.17**client**

- a) objet qui utilise les services d'un autre objet (serveur) pour exécuter une tâche
- b) initiateur d'un message auquel un serveur réagit, tel que le rôle d'une extrémité AR par laquelle il émet des APDU de demande de service confirmées à une extrémité AR unique qui agit en tant que serveur

3.5.18**voie de transport**

flux unidirectionnel d'APDU dans le cadre d'une relation d'application

3.5.19**cyclique**

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

3.5.20**AR dédiée**

AR employée directement par l'utilisateur FAL. Dans le cas des AR dédiées, seuls l'en-tête FAL et les données utilisateur sont transférés

3.5.21**dispositif**

connexion matérielle physique à la liaison. Un dispositif peut contenir deux nœuds ou plus

3.5.22

profil de dispositif

ensemble d'informations et de fonctionnalités dépendantes du dispositif assurant une cohérence entre des dispositifs similaires du même type

3.5.23

extrémité

une des entités de communication impliquées dans une connexion

3.5.24

erreur

différence entre une valeur ou une condition calculée, observée ou mesurée et la valeur ou la condition spécifiée ou théoriquement correcte

3.5.25

code d'erreur

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

3.5.26

information de gestion

information accessible sur le réseau qui vient à l'appui de la gestion du fonctionnement du système de bus de terrain, y compris la couche application. La gestion comprend les fonctions telles que le contrôle, la surveillance et le diagnostic

3.5.27

réseau

série de nœuds reliés par un certain type de support de communication. Les voies de connexion entre toute paire de nœuds peuvent inclure des répéteurs, des routeurs et des passerelles

3.5.28

extrémité de AR prédéfinie

extrémité de AR définie localement dans un dispositif sans recourir au service de création. Les AR prédéfinies qui ne sont pas préétablies le sont avant leur utilisation

3.5.29

extrémité de AR préétablie

extrémité de AR dans un état établi au cours de la configuration des AE qui contrôlent ses extrémités

3.5.30

serveur

a) rôle d'une AREP par lequel elle renvoie une APDU de réponse de service confirmée au client à l'origine de la demande

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

3.5.31

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. 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

3.6 Définitions spécifiques à la couche application de bus de terrain

Des termes supplémentaires sont définis pour la présente partie.

3.6.1 voie analogique

signal électrique à variation continue permettant de relier un dispositif de terrain au reste du système d'acquisition ou de contrôle des données. Certains dispositifs de terrain prennent en charge plusieurs voies analogiques (entrée ou sortie). Chaque voie analogique transmet une seule variable dynamique vers ou en provenance du dispositif de terrain

3.6.2 adresse de diffusion

adresse utilisée par un maître pour transmettre une commande à tous les dispositifs. La longueur de l'adresse de diffusion est de cinq octets, sa valeur étant zéro dans tous les cas

3.6.3 occupé

dispositif occupé et ne pouvant pas exécuter une commande au moment indiqué. Un dispositif indique l'état "occupé" en renvoyant le code de réponse 32 lorsque la spécification de la commande le permet. La commande demandée n'est pas exécutée si une réponse "occupé" est renvoyée

3.6.4 ID de dispositif

indique le numéro de série du dispositif. Le fabricant doit affecter une valeur unique pour chaque dispositif dont les valeurs sont identiques pour l'ID du fabricant et le type de dispositif

3.6.5 type de dispositif

indique le type de fabrication du dispositif, 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 le dispositif. Le fabricant doit attribuer une valeur unique à chaque type de dispositif

3.6.6 variable de dispositif

élément de donnée à définition unique dans un dispositif de terrain, qui est toujours associé aux informations de processus cyclique. Une valeur de variable de dispositif varie suite à des modifications et des variations du processus auquel le dispositif est relié

3.6.7 variable dynamique

connexion entre le processus et une voie analogique. Un dispositif peut contenir des variables primaires, secondaires, tertiaires et quaternaires. Ces variables sont désignées collectivement par le terme "dynamiques"

3.6.8 étiquette de grande longueur

chaîne de 32 caractères ISO à alphabet latin n°1 utilisée pour identifier un dispositif de terrain

3.6.9 courant de boucle

valeur mesurée par un milliampèremètre monté en série avec le dispositif de terrain. Le courant de boucle est un signal proche analogique en c.c. de 4-20 mA utilisé pour communiquer une valeur unique entre le système de commande et le dispositif de terrain. Les dispositifs en mode tension utilisent l'expression "Volts c.c." comme unités techniques lorsque des valeurs de "courant de boucle" sont utilisées

3.6.10 ID du fabricant

désigne le fabricant ayant produit le dispositif. Un fabricant doit utiliser la valeur qui lui est attribuée et ne doit pas utiliser la valeur attribuée à un autre fabricant

3.6.11 maître

dispositif qui initie une activité de communication par la transmission d'une PDU de demande à un dispositif

3.6.12 adresse d'interrogation

entier utilisé pour identifier le dispositif. L'adresse d'interrogation permet de constituer une adresse d'une longueur de un octet

3.6.13 esclave

dispositif qui initie une activité de communication uniquement après réception d'une PDU de demande émise par un dispositif maître et qui doit nécessairement envoyer une réponse à cette demande

3.6.14 étiquette

chaîne de 8 caractères ASCII utilisée pour identifier le dispositif de terrain

3.6.15 adresse unique

adresse d'un dispositif d'une longueur de cinq octets qui identifie de manière unique le dispositif parmi tous les autres dispositifs qui viennent à l'appui de la présente norme. Elle constitue une concaténation de l'ID du fabricant, du type de dispositif et de l'ID du dispositif

3.7 Abréviations et symboles

AE	Entité d'application
AL	Couche application
ALP	Protocole de couche application
APO	Objet d'application
AP	Processus d'application
APDU	Unité de données de protocole d'application
API	Identifiant de processus d'application
AR	Relation d'application
AREP	Extrémité de relation d'application
ASCII	Code américain normalisé pour l'échange d'information
ASE	Élément de service d'application
Cnf	Confirmation
DL	Liaison de données (comme préfixe)
DLC	Connexion de liaison de données
DLL	Couche de liaison de données
DLM	Gestion de liaison de données
DLSAP	Point d'accès au service de liaison de données
DLSDU	Unité de données de service DL
FAL	Couche application de bus de terrain
ID	Identifiant
CEI	Commission électrotechnique internationale
Ind	Indication

OSI	Interconnexion de systèmes ouverts
Req	Demande
Rsp	Réponse
SAP	Point d'accès au service
SDU	Unité de données de service
VFD	Dispositif de terrain virtuel

3.8 Conventions

3.8.1 Vue d'ensemble

La FAL est définie comme un ensemble d'ASE orientées objet. Chaque ASE est spécifiée 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.

3.8.2 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 la 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 sa 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 présente norme réserve les identifiants de classe compris entre 1 et 255 à l'identification des classes normalisées. Ces identifiants ont été affectés afin de maintenir la compatibilité avec les normes nationales existantes. Les CLASS ID compris entre 256 et 2048 sont affectés pour l'identification des classes définies par l'utilisateur.
- (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 doivent par conséquent pas être redéfinis 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 la présente norme.

- (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 étiquette 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 utilisé 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 doit être sélectionné lorsqu'une instance de la classe est définie.
 - b) L'étiquette "OpsService" désigne un service opérationnel (1).
 - b) L'étiquette "MgtService" désigne un service de gestion (2).

- d) Le numéro de ligne définit la séquence et le niveau d'imbrication de la ligne. Chaque niveau d'imbrication est identifié par période. L'imbrication dans la liste des services permet de spécifier les services soumis à une déclaration de contrainte.

3.8.3 Conventions pour les définitions de services

3.8.3.1 Généralités

La présente norme 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.

3.8.3.2 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.

NOTE 1 Voir la note en 3.8.3.3 relative à la non inclusion des paramètres de services appropriés à une spécification de protocole, d'interface de programmation ou de mise en œuvre, mais ne se rapportant toutefois pas à la définition d'un service abstrait.

La présente norme 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 de la présente norme. 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 2 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 3 Les primitives de demande, indication, réponse et confirmation sont également connues comme primitives *requestor.submit* (soumission par le demandeur), *acceptor.deliver* (remise à l'accepteur), *acceptor.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.
- (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.

3.8.3.3 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 à contrainte de temps dans la couche application de bus de terrain.

NOTE La sous-série de normes CEI 61185-5 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 une norme de spécification de protocole de la sous-série CEI 61158-6, mais non dans une norme 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 appairer 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 une norme de service abstrait ou à une norme de protocole, sauf à un niveau d'abstraction nécessairement commun à toutes les représentations de la norme 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 dans la CEI/TR 61158-1:2010, 9.6.

4 Concepts

Les concepts et modèles communs utilisés pour décrire le service de la couche application dans la présente norme sont détaillés à l'Article 9 de la IEC/TR 61158-1.

5 ASE des types de données

5.1 Vue d'ensemble

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. La présente norme 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.

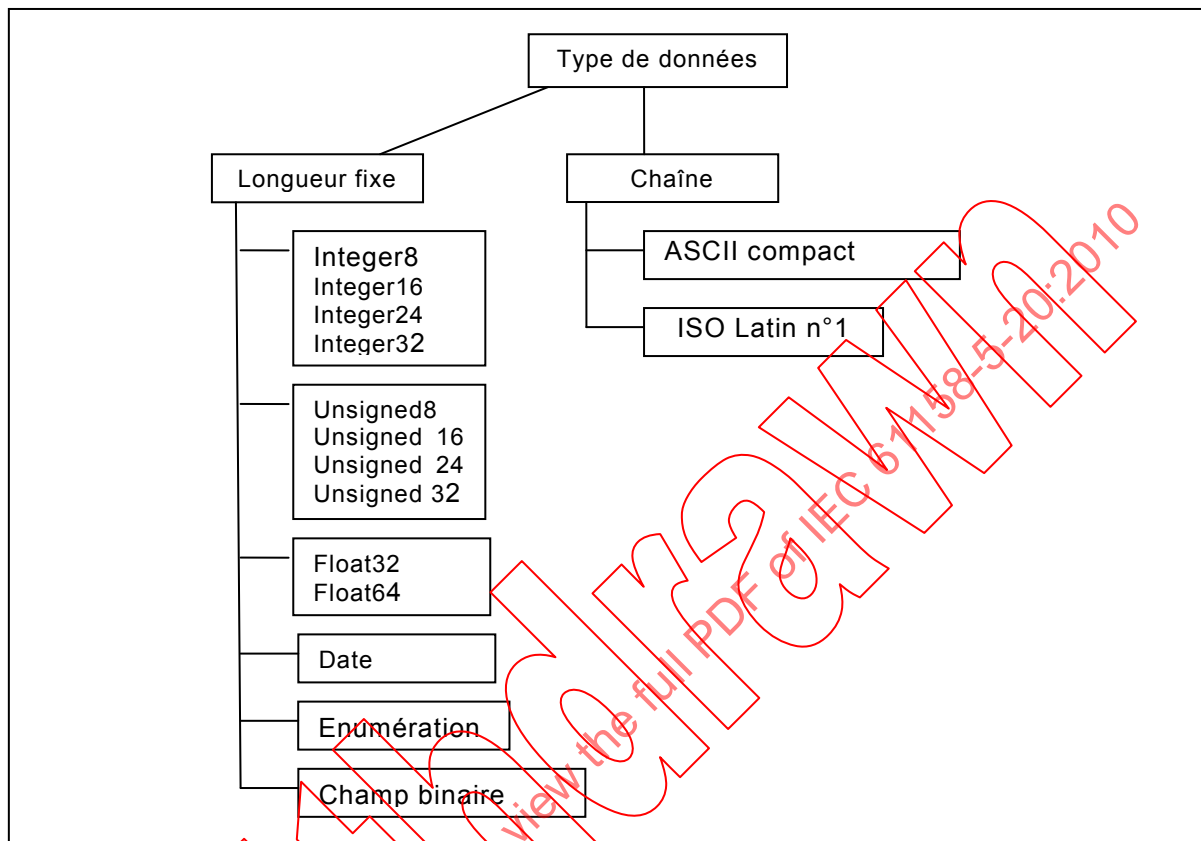


Figure 1 – Hiérarchie des classes de types de données

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 permettent de définir des types de données de longueur fixe. Les types communs issus de l'ISO/CEI 8824 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 la présente norme sont des chaînes, des matrices et des structures. Aucun type commun n'est défini pour les matrices et les structures.

5.1.1 Vue d'ensemble des types de base

La plupart des types de base sont définis à partir d'un ensemble de types définis dans l'ISO/CEI 8824 (types simples). Certains types définis dans ladite norme 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. Ils sont définis dans la présente norme 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 La présente norme définit les éléments Entier, ASCII compact, ISO Latin n°1 et Date afin de leur affecter des identifiants de classe de bus de terrain. La présente norme ne modifie pas leurs définitions spécifiées dans l'ISO/CEI 8824.

5.1.1.1 Types de longueur fixe

La longueur des types de longueur fixe est un entier relatif d'octets.

5.1.2 Vue d'ensemble des types construits

Les types de données construits sont nécessaires pour l'acheminement complet des diverses informations présentes sur le bus de terrain. Il existe deux types de données construits définis pour la présente norme, à savoir les matrices et les structures.

5.1.2.1 Chaînes

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

5.1.3 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. La présente norme ne prend pas en charge les types définis par l'utilisateur.

5.1.4 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	(o) Attribut clé:	identifiant numérique du type de données	
2	(m) Attribut clé:	Nom de type de données	
3	(m) Attribut:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)	
4	(c) Contrainte	Format = FIXED LENGTH STRING	
4,1	(m) Attribut:	Longueur en octets	
5	(c) Contrainte:	Format = STRUCTURE	
5,1	(m) Attribut:	Nombre de champs	
5,2	(m) Attribut:	Liste de champs	
5.2.1	(o) Attribut:	Nom de champ	

5.2.2	(m)	Attribut:	Type de données de champ
6	(c)	Contrainte:	Format = ARRAY
6.1	(m)	Attribut:	Nombre d'éléments de matrice
6.2	(m)	Attribut:	Type de données d'éléments de matrice

5.2.1.1 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'identification du champ par numéro permet de prendre en charge l'accès partiel aux champs d'une structure. Cet attribut est présent lorsque la valeur de l'attribut de format est "STRUCTURE".

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 par son identifiant numérique, soit par intégration de la définition d'un type de données construit à ce niveau. Lors de l'intégration d'une définition, la description du type de données intégré présentée ci-dessous est utilisée.

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 soit par référencement de la définition d'un type de données construit par son identifiant numérique, soit par intégration de la définition d'un type de données construit à ce niveau.

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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Integer8
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Integer16
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Integer24
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Integer32
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Integer8
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Unsigned16
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Unsigned24
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Unsigned32
- 3 Format = FIXED LENGTH
- 4.1 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 Identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Float32
- 4 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Float64
- 4 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Date
- 4 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Enumération
- 3 Format = FIXED LENGTH
- 4.1 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 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Champ binaire
- 3 Format = FIXED LENGTH
- 5.1 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.

5.3.2 Types de chaînes

5.3.2.1 ASCII compact

CLASS: Data type

ATTRIBUTES:

- 1 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = ASCII compact
- 3 Format = STRING
- 4.1 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	:	;	<	=	>	?

^a SP indique un caractère espace.

NOTE Le chiffre hexadécimal de poids fort se lit de haut en bas; le chiffre de poids faible se lit de gauche à droite.

5.3.2.2 ISO Latin n°1**CLASS:****Data type****ATTRIBUTES:**

- 1 identifiant numérique de type de données = Non utilisé
- 2 Nom de type de données = Latin n°1
- 3 Format = STRING
- 4.1 Longueur en octets = 1 à n

Ce type est un sous-ensemble du type de chaîne de l'ISO 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	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
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	ß

^a SP indique un caractère espace.

^b NBSP indique un caractère espace continu.

^c SHY indique un tiret.

NOTE Le chiffre hexadécimal de poids fort se lit de haut en bas; le chiffre de poids faible se lit de gauche à droite. Les cellules grisées indiquent qu'aucun caractère n'est affecté à ce code.

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	Énumération	5.3.1.12
Unsigned8	5.3.1.5	Champ binaire	5.3.1.13
Unsigned16	5.3.1.6	ASCII compact	5.3.2.1
Unsigned24	5.3.1.7	ISO Latin n°1	5.3.2.2
Unsigned32	5.3.1.8		

6 Spécification du modèle de communication

6.1 Paramètres communs

De nombreux services ont les paramètres suivants. Les définitions communes suivantes sont fournies en lieu et place d'une définition pour chaque service.

AREP ID

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

État de communication

Ce paramètre spécifie les informations concernant la défaillance de communication. Les valeurs possibles de ce paramètre sont présentées dans le Tableau 4.

Tableau 4 – Valeurs d'état de communication

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

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 le dispositif. Les valeurs possibles de ce paramètre sont présentées dans le Tableau 5.

Tableau 5 – 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; le dispositif est en mode protection d'écriture.)

Etat du dispositif

Ce paramètre indique l'état de fonctionnement courant du dispositif dans son ensemble et n'est pas associé à l'exécution d'une commande. Les valeurs possibles de ce paramètre sont présentées dans le Tableau 6.

Tableau 6 – Valeurs d'état du dispositif

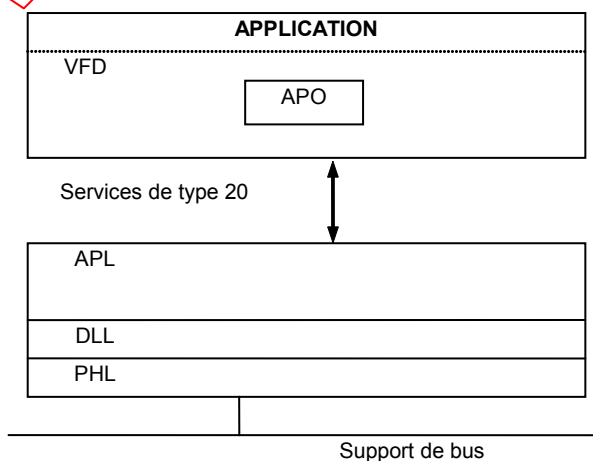
Valeur	Description
Dysfonctionnement du dispositif	Le dispositif a détecté une erreur ou une défaillance grave qui remet en cause son fonctionnement.
Modification de la configuration	Une opération de modification de la configuration du dispositif a été effectuée.
Démarrage à froid	Une panne d'alimentation ou une réinitialisation du dispositif s'est produite.
Plus grand nombre d'informations sur l'état disponibles	Un plus grand nombre d'informations sur l'état sont disponibles via une autre commande Lecture.
Fixation du courant de boucle	Le courant de boucle est maintenu à une valeur fixe et ne répond pas à des variations de processus.
Saturation du courant de boucle	Le courant de boucle a atteint sa limite d'extrémité supérieure (ou inférieure) et ne peut pas augmenter (ou diminuer) davantage.
Variable non primaire hors limites	Une variable de dispositif non mise en correspondance avec la variable primaire est supérieure à ses limites de fonctionnement.
Variable primaire hors limites	La variable primaire est supérieure à ses limites de fonctionnement.

6.2 ASE

6.2.1 ASE du dispositif de terrain virtuel

6.2.1.1 Spécification des classes de dispositif de terrain virtuel

Le dispositif 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 de la présente norme. Il convient d'indiquer uniquement de quelle manière les services décrits de façon abstraite peuvent être mis à disposition de l'application.

Un dispositif 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 de dispositif étendu
4. (m) Attribut:	ID de dispositif
5. (m) Attribut:	Rév. de dispositif
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 de dispositif
11. (m) Attribut:	Rev. de commande
12. (m) Attribut:	Compte de variables
13. (m) Attribut:	Compteur de modification de configuration
14. (m) Attribut:	ExtStatus de dispositif
15. (m) Attribut:	Code de diffusion
16. (m) Attribut:	Profil de dispositif
SERVICES:	
1. (m) Service Ops:	Identification

6.2.1.2.1 Attributs

ID du fabricant

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

Type de dispositif étendu

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

ID de dispositif

Cet attribut indique un numéro de série du dispositif. Le fabricant doit affecter une valeur unique pour chaque dispositif dont les valeurs sont identiques pour l'ID du fabricant et le type de dispositif.

Rév. de dispositif

Cet attribut décrit le niveau de révision du dispositif. 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 le dispositif.

Rév. logicielle

Cet attribut décrit le niveau de révision du micrologiciel du dispositif. Le fabricant doit augmenter la valeur de cet attribut pour chaque nouvelle version du micrologiciel du dispositif.

Rév. matérielle

Cet attribut indique le niveau de révision du matériel du dispositif. Le fabricant doit augmenter la valeur de cet attribut pour chaque modification importante du micrologiciel du dispositif. 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 le dispositif.

Compte des préambules

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

Drapeau de dispositif

Cet attribut indique d'autres informations concernant le dispositif, telles que dispositif 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 le dispositif.

Compte de variables

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

Compteur de modification de configuration

Cet attribut comptabilise le nombre de modifications de configuration du dispositif. Le dispositif doit 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 du dispositif modifie la configuration du dispositif en appliquant des moyens locaux tels que l'interface locale de l'opérateur.

ExtdStatus de dispositif

Cet attribut indique l'état opérationnel étendu du dispositif.

Code de diffusion

Cet attribut désigne le fabricant de marque privé ayant diffusé le dispositif.

Profil de dispositif

Cet attribut spécifie la classe d'appartenance du dispositif.

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 du dispositif.

6.2.1.3.2 Primitives de service

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

Tableau 7 – Paramètres de service d'identification

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

Etiquette

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