IEC 61158-5-17

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 5-17: Application layer service definition – Type 17 elements

IEC 61158-5-17:2007(E)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland
Email: inmail@iec.ch
Web: 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.

- Catalogue of IEC publications: www.iec.ch/searchpub

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub

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

- Electropedia: www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 20 000 terms and definitions in English and French, with equivalent terms in additional languages. Also known as the International Electrotechnical Vocabulary online.

- Customer Service Centre: www.iec.ch/webstore/custserv

If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

Email: csc@iec.ch
Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00

IEC 61158-5-17

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 5-17: Application layer service definition – Type 17 elements

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE XA

ICS 35.100.70; 25.040.40

ISBN 2-8318-9469-7

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-17: Application layer service definition – Type 17 elements

FOREWORD

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

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission of their respective intellectual-property-right holders.

International Standard IEC 61158-5-17 has been prepared by subcommittee 65C: Digital communications, of IEC technical committee 65: Industrial-process measurement and control.

This first edition and its companion parts of the IEC 61158-5 subseries cancel and replace IEC 61158-5:2003. This edition of this part constitutes a technical addition. This part and its Type 17 companion parts also cancel and replace IEC/PAS 62405.

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

a) deletion of the former Type 6 fieldbus for lack of market relevance;

b) addition of new types of fieldbuses;

c) partition of part 5 of the third edition into multiple parts numbered -5-2, -5-3, ...

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

FDIS	Report on voting
65C/475/FDIS	65C/486/RVD

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

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

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

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

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

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

INTRODUCTION

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

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-17: Application layer service definition – Type 17 elements

1 Scope

1.1 Overview

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

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 17 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

1) the FAL user at the boundary between the user and the Application Layer of the Fieldbus Reference Model, and

2) Systems Management at the boundary between the Application Layer and Systems Management of the Fieldbus Reference Model.

This standard specifies the structure and services of the 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 17 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 all other undated references, the latest edition of the referenced document (including any amendments) applies.

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

ISO/IEC 7498 (all parts), *Information technology – Open Systems Interconnection – Basic Reference Model*

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 Definitions

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

3.1 Terms and definitions

3.1.1 ISO/IEC 7498-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 7498-1 apply:

a) application entity

b) application protocol data unit

c) application service element

3.1.2 ISO/IEC 10731 terms

a) (N)-connection

b) (N)-entity

c) (N)-layer

d) (N)-service

e) (N)-service-access-point

3.1.3 Other terms and definitions

3.1.3.1
application
function or data structure for which data is consumed or produced

3.1.3.2
application process
part of a distributed application on a network, which is located on one device and unambiguously addressed

3.1.3.3
application relationship
cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation

NOTE This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities

3.1.3.4
application relationship endpoint
context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

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

3.1.3.5
attribute
description of an externally visible characteristic or feature of an object

NOTE 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.

3.1.3.6
behaviour
indication of how an object responds to particular eventss

3.1.3.7
bridge
intermediate equipment that connects two or more segments using a data-link layer relay function

3.1.3.8
channel
single physical or logical link of an input or output application object of a server to the process

3.1.3.9
class
a set of objects, all of which represent the same kind of system component

NOTE A class is a generalisation of an 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.

3.1.3.10
client

a) object which uses the services of another (server) object to perform a task

b) initiator of a message to which a server reacts

3.1.3.11
connection
logical binding between application objects that may be within the same or different devices

NOTE 1 Connections may be either point-to-point or multipoint.

NOTE 2 The logical link between sink and source of attributes and services at different custom interfaces of RT-Auto ASEs is referred to as interconnection. There is a distinction between data and event interconnections. The logical link and the data flow between sink and source of automation data items is referred to as data interconnection. The logical link and the data flow between sink (method) and source (event) of operational services is referred to as event interconnection.

3.1.3.12
connection point
buffer which is represented as a subinstance of an Assembly object

3.1.3.13
conveyance path
unidirectional flow of APDUs across an application relationship

3.1.3.14
dedicated AR
AR used directly by the FAL User

NOTE On Dedicated ARs, only the FAL Header and the user data are transferred.

3.1.3.15
device
physical hardware connected to the link

NOTE A device may contain more than one node.

3.1.3.16
domain
part of the RTE network consisting of one or two subnetwork(s)

NOTE Two subnetworks are required to compose a dual-redundant RTE network, and each end node in the domain is connected to both of the subnetworks.

3.1.3.17
domain master
station which performs diagnosis of routes to all other domains, distribution of network time to nodes inside the domain, acquisition of absolute time from the network time master and notification of status of the domain

3.1.3.18
domain number
numeric identifier which indicates a domain

3.1.3.19
end node
producing or consuming node

3.1.3.20
endpoint
one of the communicating entities involved in a connection

3.1.3.21
error
discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.1.3.22
external bridge
bridge to which neither internal bridges nor RTE stations are connected directly

3.1.3.23
event
an instance of a change of conditions

3.1.3.24
group

a) <general> a general term for a collection of objects. Specific uses:

b) <addressing> when describing an address, an address that identifies more than one entity

3.1.3.25
interface

a) shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate

b) collection of FAL class attributes and services that represents a specific view on the FAL class

3.1.3.26
interface port
physical connection point of an end node, which has an independent DL-address

3.1.3.27
internal bridge
bridge to which no routers, external bridges or nodes non-compliant with this specification are connected directly

3.1.3.28
invocation
act of using a service or other resource of an application process

NOTE Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation.

3.1.3.29
junction bridge
bridge to which at least one router, external bridge or node non-compliant with this specification, and to which at least one internal bridge or RTE station is connected

3.1.3.30
link
physical communication channel between two nodes

3.1.3.31
method
<object> a synonym for an operational service which is provided by the server ASE and invoked by a client

3.1.3.32
network
a set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.1.3.33
network time master
station which distributes network time to domain masters

3.1.3.34
node
single DL-entity as it appears on one local link

3.1.3.35
non-redundant interface node
node whch has a single interface port

3.1.3.36
non-redundant station
station that consists of a single end node

NOTE “non-redundant station” is synonymous with “end node”.

3.1.3.37
object
abstract representation of a particular component within a device, usually a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behaviour

3.1.3.38
originator
client responsible for establishing a connection path to the target

3.1.3.39
path
logical communication channel between two nodes, which consists of one or two link(s)

3.1.3.40
peer
role of an AR endpoint in which it is capable of acting as both client and server

3.1.3.41
producer
node that is responsible for sending data

3.1.3.42
provider
source of a data connection

3.1.3.43
publisher
role of an AR endpoint that transmits APDUs onto the fieldbus for consumption by one or more subscribers

NOTE A publisher may not be aware of the identity or the number of subscribers and it may publish its APDUs using a dedicated AR.

3.1.3.44
redundant interface node
node with two interface ports one of which is connected to a primary network, while the other is connected to a secondary network

3.1.3.45
redundant station
station that consists of a pair of end nodes

NOTE Each end node of a redundant station has the same station number, but has a different DL-address.

3.1.3.46
resource
a processing or information capability of a subsystem

3.1.3.47
RTE station
station compliant with this specification

3.1.3.48
route
logical communication channel between two communication end nodes

3.1.3.49
router
intermediate equipment that connects two or more subnetworks using a network layer relay function

3.1.3.50
segment
communication channel that connects two nodes directly without intervening bridges

3.1.3.51
server

a) role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

b) object which provides services to another (client) object

3.1.3.52
service
operation or function than an object and/or object class performs upon request from another object and/or object class

3.1.3.53
station
end node or a pair of end nodes that perform a specific application function

3.1.3.54
station number
numeric identifier which indicates a RTE station

3.1.3.55
subnetwork
part of a network that does not contain any routers. A subnetwork consists of end nodes, bridges and segments

NOTE Every end node included in a subnetwork has the same IP network address.

3.1.3.56
subscriber
role of an AREP in which it receives APDUs produced by a publisher

3.2 Abbreviations and symbols

3.2.1 ISO/IEC 10731 abbreviations

ASE	application-service-element
OSI	Open Systems Interconnection

3.2.2 Other abbreviations and symbols

AE	Application entity
AL	Application layer
AP	Application process
APDU	Application protocol data unit
APO	Application object
AR	Application relationship
AREP	Application relationship end ooint
Cnf	Confirmation
DL-	data-link layer (as a prefix)
DLL	DL-layer
DLSAP	DL-service-access-point
FAL	Fieldbus application layer
FIFO	First-in first-out (queuing method)
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
ISO	International Organization for Standardization
MSU-AR	Multipoint network-scheduled unconfirmed publisher/subscriber AREP
MTU-AR	Multipoint user-triggered unconfirmed publisher/subscriber AREP
PDU	Protocol Data Unit
PSU-AR	Point-to-point network-scheduled unconfirmed client/server AREP
PTC-AR	Point-to-point user-triggered confirmed client/server AREP
PTU-AR	Point-to-point user-triggered unconfirmed client/server AREP
Req	Request
Rsp	Response
SAP	Service Access Point

3.3 Conventions

3.3.1 Overview

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

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

3.3.2 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.

3.3.3 Conventions for service definitions

3.3.3.1 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.

3.3.3.2 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 3.3.3.3 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.

3.3.3.3 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 series 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. Protocol aspects that can vary among different protocol specifications or different implementations that instantiate the same abstract services are unsuitable for inclusion in these service definitions, except at the level of abstraction that is necessarily common to all such expressions.

For example, the means by which service providers pair request and reply PDUs is appropriate for specification in an IEC 61158-6 protocol specification standard but not in an IEC 61158-5 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 abstract service definitions can be found in IEC/TR 61158-1 (Ed.2.0), 9.6.

4 Concepts

4.1 General

This Fieldbus Application Layer (FAL) provides communication services to time-critical and non-time-critical applications in fieldbus devices.

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

4.2 Relationships between ASEs

4.3 FAL ASEs

A modular approach was taken in the definition of FAL ASEs. The ASEs defined for the FAL are also object-oriented. In general, ASEs provide a set of services designed for one specific object class or for a related set of classes.

To support remote access to the AP, the Application Relationship ASE is defined. This application relationship ASE provides services to the AP for defining and establishing communication relationships with other APs, and provides services to other ASEs for conveying their service requests and responses.

Each FAL ASE defines a set of services, APDUs and procedures that operate on the classes that it represents. Only a subset of the ASE services may be provided to meet the needs of an application. Profiles may be used to define such subsets.

APDUs are sent and received between FAL ASEs that support the same services. Figure 1 shows the FAL ASEs from the perspective of communication.

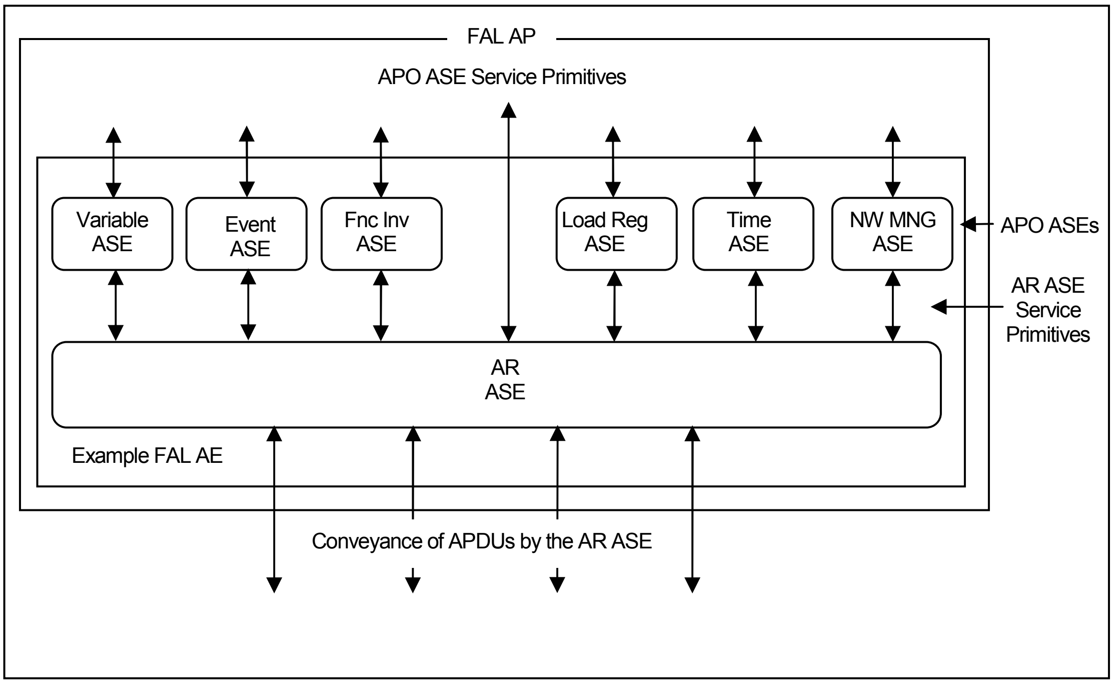

Figure 1 – FAL ASEs

4.4 Common FAL service parameters

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

Argument
This parameter carries the parameters of the service invocation.

AREP
This parameter specifies information sufficient for local identification of the AREP to be used to convey the service. This parameter may use a key attribute of the AREP to identify the application relationship. When an AREP supports multiple contexts simultaneously, the AREP parameter is extended to identify the context as well as the AREP.

Result(+)
This selection type parameter indicates that the service request was successful.

Result(–)
This selection type parameter indicates that the request failed.

Error info
This parameter provides error information for service errors. It is returned in confirmed service response(-) primitives.

5 ASEs

5.1 Variable ASE

5.1.1 Overview

In the fieldbus environment, application processes contain data that can be both read and written by remote applications. The variable ASE defines the network-visible attributes of application data and provides a set of services used to read, write and report their values.

When the appropriate types of application relationship are supported, the variable ASE may be used to support two different access models i.e., the client/server model and the publisher/subscriber model. The client/server model is characterized by a client application sending a read or write request to a server application that responds accordingly. The server's activity is stimulated by clients on the network; if there are no requests, the server generates no responses.

The publisher/subscriber model is different in that it is characterized by a data producer publishing its data onto the network. Subscribers wishing to acquire the published data join the application relationship used to publish it and listen for the data as it is transmitted. Two models are provided to support this publisher/subscriber activity, i.e., the “pull” model and the “push” model.

In the “pull” model, one of the subscribers requests that the publisher publishes a sequence of variable data by issuing a publish request to it. The publisher distributes the variable data periodically according to the remote request by multicasting. The publishing schedule is controlled by the publisher itself.

In the “push” model, the publisher is requested to distribute a sequence of variable data by the local FAL user. The publisher distributes a sequence of variable data by multicasting according to the local request. The publishing schedule is also controlled by the publisher itself.

5.1.2 Variable class specification

5.1.2.1 Formal model

FAL ASE:			**VARIABLE ASE**
CLASS:			**VARIABLE**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Key Attribute:	Numeric Identifier
2	(m)	Attribute:	Data type
3	(0)	Attribute:	Length
4	(o)	Attribute:	Read enable
5	(o)	Attribute:	Write enable
SERVICES:			
1	(o)	OpsService:	Read
2	(o)	OpsService:	Write
3	(o)	OpsService:	Information Report

5.1.2.2 Attributes

Numeric identifier
This key attribute identifies an instance of this object class.

Data type
This attribute is the numeric identifier of the data type.

Length
This optional attribute is the length of the data in octets.

Read enable
The value of this optional attribute indicates whether the data value of the Variable Object can be read via the fieldbus.

Write enable
The value of this optional attribute indicates whether the data value of the Variable Object can be updated via the fieldbus.

5.1.3 Variable List class specification

5.1.3.1 Formal model

FAL ASE:			**VARIABLE ASE**
CLASS:			**VARIABLE LIST**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Key Attribute:	Numeric Identifier
2	(m)	Attribute:	Number of Entries
3	(m)	Attribute:	List Of Variables
SERVICES:			
1	(o)	OpsService:	Read
2	(o)	OpsService:	Write
3	(o)	OpsService:	Information Report

5.1.3.2 Attributes

Numeric identifier
This key attribute identifies an instance of this object class.

Number of entries
This attribute specifies the number of variables in the list.

List of variables
This attribute identifies the variables by the key attributes that are contained in the list.

5.1.4 Variable ASE service specification

5.1.4.1 Supported services

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

- Read
- Write
- Information Report

5.1.4.2 Read service

5.1.4.2.1 Service overview

This confirmed service may be used to read the value of a variable object or a variable list object. It is not used with the publisher/subscriber model.

5.1.4.2.2 Service primitives

The service parameters for each primitive are shown in Table 1.

Table 1 – Read service parameters

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

Variable specifier
This parameter specifies a variable object or a variable list object to be read by the key attribute.

Value
This parameter specifies the value read. For each of the variables, this parameter specifies the value of the variable. For variable lists, this parameter specifies the values of each of the variables in the list concatenated together in the order in which they appear in the list.

NOTE If any of the variables in a variable list could not be read, the service fails.

5.1.4.3 Write service

5.1.4.3.1 Service overview

This confirmed service is used to write the value of a variable object or a variable list object. It is not used with the publisher/subscriber model.

5.1.4.3.2 Service primitives

The service parameters for each primitive are shown in Table 2.

Table 2 – Write service parameters

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

Variable specifier
This parameter specifies a variable object or a variable list object to be written by the key attribute.

Value
This parameter specifies the value to write. For variables, this parameter specifies the value of the variable. For variable lists, this parameter specifies the values of each of the variables in the list concatenated together in the order that they appear in the list.

NOTE If any variable in a variable list object cannot be updated, none of the variables in the variable list object will be updated and the write will fail.

5.1.4.4 Information report service

5.1.4.4.1 Service overview

This optional service is an unconfirmed service that may be used to report the value of a variable object or a variable list object. This service may be used in the publisher/subscriber push model.

5.1.4.4.2 Service primitives

The service parameters for each primitive are shown in Table 3.

Table 3 – Information report service parameters

Parameter name	Req	Ind
Argument		
AREP	M	M
Variable Specifier	M	M (=)
Value	M	M (=)

Variable specifier
This parameter specifies a variable object or a variable list object to be reported by the key attribute.

Value
This parameter specifies the value to be reported. For variables, this parameter specifies the value of the variable. For variable lists, this parameter specifies the value of each variable in the list concatenated together in the order that they appear in the list.

NOTE If any of the variables in a variable list could not be read, the service fails.

5.2 Event ASE

5.2.1 Overview

Event objects are used to define messages reporting event occurrences. Event messages contain information that identifies and describes event occurrences.

Notifiers are responsible for collecting event messages from event objects, and distributing one or more event message(s) in a single invocation of the FAL event notification service. The number of event messages that may be submitted in a single service invocation is limited by the maximum APDU size that can be transferred by the AR.

If an application process fails to receive one or more event notifications, a notification recovery service is provided to request the notifier to retransmit the notifications.

In this model, application processes are responsible for providing the functions for event objects and event list objects, and the FAL is responsible for providing communication services designed specifically for them. The application process detects events, builds event messages and aggregates them together. The application process distributes the aggregated event messages using the FAL event notification service.

5.2.2 Event class specification

5.2.2.1 Formal model

FAL ASE:			**EVENT ASE**
CLASS:			**NOTIFIER**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Key Attribute:	Numeric Identifier
2	(m)	Attribute:	AREP
3	(o)	Attribute:	Last Notification Sequence Number
4	(o)	Attribute:	List Of Events
SERVICES:			
1	(o)	OpsService:	Event Notification
2	(o)	OpsService:	Notification Recovery

5.2.2.2 Attributes

5.2.2.2.1 Numeric identifier

This key attribute identifies an instance of this object class.

5.2.2.2.2 AREP

This attribute identifies the AREP configured to convey event notifications. This AREP is also used for reporting the event notifications generated by an event recovery request.

5.2.2.2.3 Last notification sequence number

The conditional attribute specifies the last sequence number used. It is incremented for each event notification service invocation.

5.2.2.2.4 List of events

This optional attribute identifies the events that are configured.

5.2.3 Event ASE service specification

5.2.3.1 Supported services

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

- Event Notification
- Notification Recovery

5.2.3.2 Event notification service

5.2.3.2.1 Service overview

This unconfirmed service is used by the notifier of an FAL AP to notify other APs that one or more events have occurred.

5.2.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 4.

Table 4 – Event notification service parameters

Parameter name	Req	Ind
Argument		
AREP	M	M
NotifierID	M	M (=)
Sequence Number	U	U (=)
Notification Time	U	U (=)
List of Event Messages	M	M (=)
Event Key Attribute	M	M (=)
Event Data type	C	C (=)
Event Detection Time	U	U (=)
Event Data	U	U (=)

NotifierID
This parameter identifies the notifier issuing the event notification.

Sequence number
This optional parameter is the sequence number for the event notification. It may be used for notification recovery purposes.

Notification time
This optional parameter is the time of the event notification.

List of event messages
This parameter specifies the list of event messages that are to be reported. It may contain messages from one or more event objects. The contents of each message are specified by its event object and should be consistent with that specified for the notifier object.

Event key attribute
This parameter identifies each of the specific events being acknowledged by this service.

Event data type
This conditional parameter indicates the data type of each of the event data parameters. This parameter may be present only if the event data parameter is present. If the event data parameter is present, this parameter should be present.

Event detection time
This optional parameter reports the time of the event detection. This parameter is present only if it is defined for the specified event object and is supported by the specified notifier.

Event data
This optional parameter specifies user data to be included in an event message in addition to that used to identify the event occurrence. This parameter is present only if it is defined for the specified event object and is supported by the specified notifier.

5.2.3.3 Notification recovery service

5.2.3.3.1 Service overview

This unconfirmed service is used to request that a specified number of retained event notifications be returned. Notifications are returned using the event notification service.

5.2.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 5.

Table 5 – Event notification recovery service parameters

Parameter name	Req	Ind
Argument		
AREP	M	M
NotifierID	M	M (=)
Sequence Number	U	U (=)

NotifierID
This parameter identifies the notifier to which this service is directed.

Sequence number
This optional parameter specifies the sequence number of the event notification to be re-sent. If not present, the last notification sent is being requested.

5.3 Load region ASE

5.3.1 Overview

A load region represents an unstructured memory area whose contents is to be uploaded (read) or downloaded (written). In this context, “unstructured” means that the memory area is represented only as an ordered sequence of octets. No other structure is apparent.

A load region may represent an unnamed volatile memory area, such as that implemented by dynamic computer memory, or a named non-volatile memory object, such as a file. The contents of a load region are referred to as a load image and can contain programs or data. The transfer of a load image to or from a load region is performed using the load process.

This load region model provides services that permit an application process to initiate the downloading or uploading of specified load regions.

5.3.2 Load region class specification

5.3.2.1 Formal model

FAL ASE:			**LOAD REGION ASE**
CLASS:			**LOAD REGION**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Attribute:	Load Region Size
2	(m)	Attribute:	Local Address
3	(o)	Attribute:	Load Image Name
SERVICES:			
1	(m)	OpsService:	Download Services
2	(m)	OpsService:	Upload Services

5.3.2.2 Attributes

Load region size
This attribute specifies the maximum size of the load region in octets.

Local address
This attribute is a locally significant address of the load region.

Load image name
This optional attribute specifies the name of the load image contained in the load region.

5.3.3 Load region ASE service specification

5.3.3.1 Supported services

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

- Download
- Upload

5.3.3.2 Download service

5.3.3.2.1 Service overview

This confirmed service is used to download a load image in its request and indication primitives. The response and confirmation primitives are used to convey the success or failure of the download.

5.3.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 6.

Table 6 – Download service parameters

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

Load region
This parameter specifies the load region into which the image is to be downloaded.

Load data
This parameter specifies the data to be downloaded.

5.3.3.3 Upload service

5.3.3.3.1 Service overview

This confirmed service is used to upload a load image. Its request and indication primitives convey an upload request. The response and confirmation primitives are used to convey the load image of the load region and the result of loading.

5.3.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 7.

Table 7 – Upload service parameters

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

Load region
This parameter specifies the load region from which the image is to be uploaded.

Load data
This parameter specifies the data uploaded.

5.4 Function invocation ASE

5.4.1 Overview

The function invocation class models the state-oriented function invocation. It may be used to model software processes or user functions the operation of which may be controlled.

5.4.2 Function invocation class specification

5.4.2.1 Formal model

FAL ASE:			**FUNCTION INVOCATION ASE**
CLASS:			**FUNCTION INVOCATION**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Key Attribute:	Identifier
2	(m)	Attribute:	Function Invocation State
SERVICES:			
1	(o)	OpsService:	Start
2	(o)	OpsService:	Stop
3	(o)	OpsService:	Resume

5.4.2.2 Attributes

5.4.2.2.1 Identifier

This key attribute consists of a station identifier and a function identifier.

5.4.2.2.2 Function invocation state

This attribute indicates the current state of the function invocation. An enumerated set of values has been defined.

UNRUNNABLE This state indicates that the function invocation is not executing and can not be executed.

IDLE This state indicates that the function invocation is not executing, but is capable of being executed.

RUNNING This state indicates that the function invocation is executing.

STOPPED This state indicates that the execution of a function invocation has been suspended.

5.4.3 Function invocation ASE service specification

5.4.3.1 Supported services

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

- Start
- Stop
- Resume

5.4.3.2 Start service

5.4.3.2.1 Service overview

This confirmed service is used to start a function invocation from the initial condition.

5.4.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 8.

Table 8 – Start service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M		
FunctionID	M	M (=)		
Result				
Error Info			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

FunctionID
This parameter specifies one of the key attributes of the function invocation object.

5.4.3.3 Stop service

5.4.3.3.1 Service overview

This confirmed service is used to stop a function invocation retaining its context so that it may be resumed.

5.4.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 9.

Table 9 – Stop service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
FunctionID	M	M (=)		
Result				
Error Info			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

FunctionID
This parameter specifies one of the key attributes of the function invocation object.

5.4.3.4 Resume service

5.4.3.4.1 Service overview

This confirmed service is used to request to start a function invocation from the suspended condition.

5.4.3.4.2 Service primitives

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

Table 10 – Resume service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M (=)		
FunctionID	M	M (=)		
Result				
Error Info			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

FunctionID
This parameter specifies one of the key attributes of the function invocation object.

5.5 Time ASE

5.5.1 Overview

The Time ASE specifies the clock synchronization mechanism to synchronize the time among network stations. This time can be used to add a time value to alert information and to sequence messages in a time-wise order.

A system consists of one or more time master(s) and several time slaves. Clock synchronization is performed by the communication system. Adjustment and management of the clock are tasks of the application.

The network should have at least one network time master, which may be one of the domain time masters selected automatically or may be a fixed station.

The network time master has the master clock of the network. The master clock may be the local clock of the network time master itself or it may be synchronized with a clock source outside the network.

The domain time master of each domain obtains network time from the network time master and it distributes the network time value to stations in the domain.

5.5.2 Time class specification

5.5.2.1 Formal model

FAL ASE:			**Time ASE**
CLASS:			**Time**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Attribute:	Implicit
2	(m)	Attribute:	Role
3	(o)	Attribute:	Stratum
4	(o)	Attribute:	Poll Interval
5	(o)	Attribute:	Precision
SERVICES:			
1	(m)	OpsService:	Get Network Time
2	(m)	OpsService:	Set Network Time
3	(m)	OpsService:	Tick Notification service

5.5.2.2 Attributes

5.5.2.2.1 Role

This attribute specifies the role of the Time ASE with values defined as follows:

NETWORK TIME MASTER	A network unique end node which responds to time requests from domain masters.
DOMAIN TIME MASTER	A domain unique end node which distributes time information within the domain.
TIME SLAVE	An end node subscribes time information distributed by the domain master.

5.5.2.2.2 Stratum

This attribute indicates the stratum level of the local clock.

5.5.2.2.3 Poll interval

This attribute indicates the maximum interval between successive time messages.

5.5.2.2.4 Precision

This attribute indicates the precision level of the local clock.

5.5.3 Time ASE service specification

5.5.3.1 Supported services

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

- Get Network Time
- Set Network Time
- Tick Notification

5.5.3.2 Get network time service

5.5.3.2.1 Service overview

This local service should be used to obtain the network time value that is distributed from the network. If Time ASE is the network time master, the local time value of the master clock is returned.

5.5.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 11.

Table 11 – Get network time service parameters

Parameter name	Req	Cnf
Argument		
Result		
Network Time		M
Status		M
Error info		C

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

Network time
This parameter is the value of the network time.

Status
This parameter indicates the states of time synchronization. The possible states are;

- Not synchronized
- Synchronized to the domain time master
- Synchronized to the network time master as the domain time master
- Synchronized to an external time source as a network time master

5.5.3.3 Set network time service

5.5.3.3.1 Service overview

This confirmed service should be used to set the value of network time to the time master.

This service is available under conditions where the network time master is not synchronized to an external time source.

5.5.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 12.

Table 12 – Set network time service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
Network Time	M	M (=)		
Result				
Error Info			C	C (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Network time
This parameter is the value of the network time to be set.

5.5.3.4 Tick notification service

5.5.3.4.1 Service overview

This local service should be used to recognize tick timing synchronized to the network time. The tick interval is configurable.

5.5.3.4.2 Service primitives

The service parameters for each primitive are shown in Table 13.

Table 13 – Tick notification service parameters

Parameter name	Ind
Result	
Tick	M

Tick
This parameter indicates tick timing.

NOTE This service may be implemented by means of a hard-wired interruption.

5.6 Network management ASE

5.6.1 Overview

End nodes may have two network interfaces that provide network redundancy. These interfaces are referred to as network interface A and network interface B. The former should be connected to the primary network, while the latter should be connected to the secondary network.

Each Network Management ASE constructs two diagnostic message APDUs and sends them concurrently to network interfaces A and B. Each Network Management ASE receives pairs of APDUs from other end nodes on the network that participate in network redundancy. This information is used to determine which network interface (A or B) is to be selected to send other APDUs. This information is also used to determine whether the destination node is reachable in advance of sending an APDU.

This information is preserved in the network status table.

5.6.2 Network management class specification

5.6.2.1 Formal model

FAL ASE:			**Network Management**
CLASS:			**Network Management**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Attribute:	Number of Network Interfaces
2	(m)	Attribute:	Diagnostic Message Interval
3	(m)	Attribute:	Aging Time
4	(m)	Attribute:	List of Network Interface Status
SERVICES:			
1	(m)	OpsService:	Get Network Status
2	(m)	OpsService:	Get Station Status
3	(m)	OpsService:	Network Status Change Report
4	(m)	OpsService:	Station Status Change Report

5.6.2.2 Attributes

5.6.2.2.1 Number of network interfaces

This attribute specifies the number of network interfaces on this end node. An end node may have one or two network interfaces.

5.6.2.2.2 Diagnostic message interval

This attribute specifies the time interval between successive invocations of the diagnostic messages in milliseconds.

5.6.2.2.3 Aging time

This attribute specifies the time interval used to remove silent end nodes from the List of Network Interface Status.

5.6.2.2.4 List of network interface status

This attribute is a list of the path status representing the condition of the path to a remote station from which the Network Management ASE may be receiving successive Diagnostic Messages. The status is bi-directional for a path and indicates whether Diagnostic Messages are being received successfully at each end of the path or whether either end of the path is in a fault state.

5.6.3 Network management ASE service specification

5.6.3.1 Supported services

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

The services defined for this ASE are

- Get Network Status
- Get Station Status
- Network Status Change Report
- Station Status Change Report

5.6.3.2 Get network status service

5.6.3.2.1 Service overview

This local service is used to obtain the status of the network.

5.6.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 14.

Table 14 – Get network status service parameters

Parameter name	Req	Cnf
Argument		
Result		
Network 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.		

Network status
This parameter indicates consistency of the primary and secondary networks. Possible values are

- HEALTHY
- FAILED

5.6.3.3 Get station status service

5.6.3.3.1 Service overview

This local service is used to obtain the status of the specified station.

5.6.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 15.

Table 15 – Get station status service parameters

Parameter name	Req	Cnf
Argument		
Station Identifier	M	
Result		
Station Status		M
Route 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.		

Station identifier
This parameter indicates the station for which the status is requested, and consists of a domain number and a station number.

Station status
This parameter indicates the status of the station specified by the station identifier, and includes the following information.

a) existence

TRUE	the station exists
FALSE	the station does not exist

b) redundancy

SINGLE	the station is not redundant
DUAL-REDUNDANT	the station is redundant

c) AREP of the end node which is on service

Route status
This parameter indicates the status of route for the station, and includes the following information.

a) status of the route on the primary network

REACHABLE

UNREACHABLE

b) status of the route on the secondary network

REACHABLE

UNREACHABLE

5.6.3.4 Network status change report service

5.6.3.4.1 Service overview

This local service is used to inform of changes in network status.

5.6.3.4.2 Service primitives

The service parameters for each primitive are shown in Table 16.

Table 16 – Network status change report service parameters

Parameter name	Ind
Result	
Network Status	M

Network status
This parameter indicates consistency of the primary and secondary networks, and has the following possible values:

- HEALTHY
- FAILED

5.6.3.5 Station status change report service

5.6.3.5.1 Service overview

This local service is used to inform of changes in the station status.

5.6.3.5.2 Service primitives

The service parameters for each primitive are shown in Table 17.

Table 17 – Station status change report service parameters

Parameter name	Ind
Result	
Station Identifier	M
Station Status	M
Route Status	M

Station identifier
This parameter indicates the station of which the status has changed.

Station status
This parameter indicates the status of the end node specified in/by the request primitive and includes the following information:

a) existence

TRUE	the station exists
FALSE	the station does not exist

b) redundancy

SINGLE	the station is not redundant
DUAL-REDUNDANT	the station is redundant

c) AREP of the end node which is on service

Route status
This parameter indicates the status of the route for the station, and includes the following information:

a) status of the route on the primary network

REACHABLE

UNREACHABLE

b) status of the route on the secondary network

REACHABLE

UNREACHABLE

5.7 Application relationship ASE

5.7.1 Overview

5.7.1.1 General

In a distributed system, application processes communicate with each other by exchanging Application Layer messages across well-defined Application Layer communication channels.

These communication channels are modelled in the Fieldbus Application Layer as application relationships (ARs).

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

The messages that are conveyed by ARs are FAL service requests and responses. Each is submitted to the AR ASE for transfer by an FAL Application Service Element (ASE) that represents the class of the APO being accessed. Figure 2 illustrates this concept.

Depending on the type of AR, APDUs may be sent to one or more destination application processes connected by the AR. Other characteristics of the AR determine how APDUs are to be transferred. These characteristics are described below.

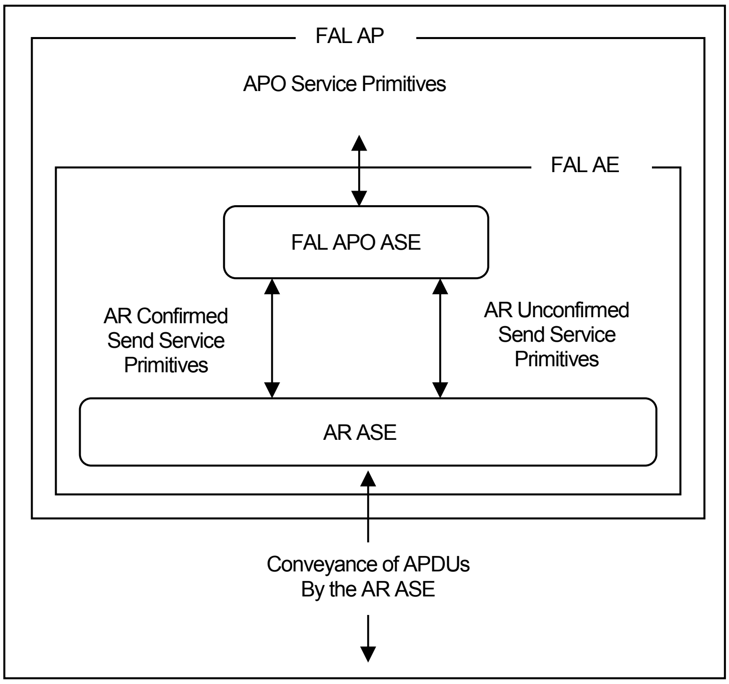
FAL AP
APO Service Primitives
FAL AE
FAL APO ASE
AR Confirmed Send Service Primitives
AR Unconfirmed Send Service Primitives
AR ASE
Conveyance of APDUs By the AR ASE

Figure 2 – The AR ASE conveys APDUs between APs

5.7.1.2 Endpoint context

Each AP involved in an AR contains an endpoint of the AR. Each AR endpoint is defined within the AE of the AP. An endpoint definition, when combined with the definitions of the other endpoints, defines an AR. To ensure communication compatibility among or between endpoints, each endpoint definition contains a set of compatibility-related characteristics.

These characteristics need to be configured appropriately for each endpoint for the AR to operate properly.

Endpoint definitions also contain a set of characteristics that describe the operation of the AR. These characteristics, when combined with those used to specify compatibility, define the context of the endpoint. The endpoint context is used by the AR ASE to manage the operation of the endpoint and the conveyance of APDUs. The characteristics that comprise the endpoint context are described in the next section.

5.7.1.2.1 Endpoint role

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

Table 18 and Table 19 summarize the characteristics and combinations of each of the AREP roles.

Table 18 – Conveyance of service primitives by AREP role

	Client	Server	Peer	Publisher	Subscriber
Send Service Req	X		X	X	
Recv Service Req		X	X		X
Send Service Rsp		X	X		
Recv Service Rsp	X		X		

Table 19 – Valid combinations of AREP roles involved in an AR

	Client	Server	Peer	Publisher	Subscriber
Client		X	X		
Server	X		X		
Peer	X	X	X		
Publisher					X
Subscriber				X	

5.7.1.2.2 Dedicated AR endpoints

Dedicated AR endpoints provide their services direct to the FAL User. Although their behaviour is the same as that of non-dedicated endpoints, the APDUs they convey contain no service-specific protocol control information other than the AR control field.

FAL Users configured for dedicated ARs construct and transfer their data without using the services of the other FAL ASEs. The format and contents of the user data conveyed over dedicated ARs are known only through configuration.

5.7.1.2.3 Cardinality

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

When expressed from the viewpoint of a client or peer endpoint, ARs are always one-to-one. Clients are never capable of issuing a request to multiple servers and waiting for responses from them.

When expressed from the viewpoint of a publisher endpoint, the cardinality indicates that multiple subscribers are supported. These one-to-many ARs provide communications between one application and a group of (one or more) applications. They are often referred to as multi-party and multicast.

In one-to-many ARs, the publisher endpoint identifies the remote endpoints using a group identifier, rather than identifying each one individually as is done with one-to-one ARs. This permits subscribers to join and leave ARs without disrupting the publisher endpoint context because their individual identities are not known to the publisher endpoint. However, the publisher application may be aware of their participation, but that information is not part of the AR endpoint context.

5.7.1.2.4 Conveyance model

The conveyance model defines how APDUs are sent between endpoints of an AR. Three characteristics are used to define these transfers:

- conveyance paths,
- trigger policy, and
- conveyance policy.

a) Conveyance paths
The purpose of AR ASEs is to transfer information between AR endpoints. This information transfer occurs over the conveyance paths of an AR. A conveyance path represents a one-way communication path used by an endpoint for input or output.

To support the role of the application process, the endpoint is configured with either one or two conveyance paths. Endpoints that only send or only receive are configured with either a send or receive conveyance path, respectively, and those that do both are configured with both. ARs with a single conveyance path are called unidirectional, and those with two conveyance paths are called bi-directional.

Unidirectional ARs are capable of conveying service requests only. To convey service responses, a bi-directional AR is necessary. Therefore, unidirectional ARs support the transfer of unconfirmed services in one direction only, while bi-directional ARs support the transfer of unconfirmed and confirmed services initiated by only one endpoint, or by both endpoints.

b) Trigger policy
Trigger policy indicates when APDUs are transmitted over the network by the data-link layer.

The first type is referred to as user-triggered. User-triggered AREPs submit FAL APDUs to the data-link layer for transmission at the earliest opportunity.

The second type is referred to as network-scheduled. Network scheduled AREPs submit FAL APDUs to the data-link layer for transmission according to a schedule configured by management.

c) Conveyance policy
Conveyance policy indicates whether APDUs are transferred according to a buffer model or a queue model. These models describe the method of conveying APDUs from the sender to the receiver.

Buffered ARs contain conveyance paths that have a single buffer at each endpoint. Updates to the source buffer are conveyed to the destination buffer according to the trigger policy of the AR. Updates to either buffer replace its contents with new data. In buffered ARs, unconveyed or undelivered data that have been replaced are lost. In addition, a data contained in a buffer may be read multiple times without destroying its contents.

Queued ARs contain conveyance paths that are represented as a queue between endpoints. Queued ARs convey data using a FIFO queue. Queued ARs are not overwritten, meaning that new entries are queued until they can be conveyed and delivered.

If a queue is full, a new message will not be placed into the queue.

5.7.1.3 AR establishment

For an AR endpoint to be used by an application process, the corresponding AR must be active. When an AR is activated, it is referred to as “established”.

AR establishment can occur in one of three ways.

First, ARs can be pre-established, meaning that the AE that maintains the endpoint context is created when the AP is connected to the network. In this case, communications among the applications involved in the AR may take place without first having to explicitly establish the AR. Any AR may be defined as being pre-established.

Second, ARs can be pre-defined, but not pre-established. Pre-defined means that the characteristics of the AR are known to each endpoint, but that their contexts have not been synchronized for operation. In this case, the use of the FAL Establish service is required to synchronize them and open them for data transfer.

Third, ARs can require dynamic definition and establishment. In this case, definitions may be created for each of the AREPs using the Create service. After creation, they are established using the Establish service if they were not created as “established”.

5.7.2 Application relationship class specification

5.7.2.1 Formal model

The AR endpoint formal model defines the characteristics common to all AR endpoints. This class is not capable of being instantiated. It is present only for the inheritance of its attributes and services by its subclasses, each specified in a separate subclause of this standard.

All AR endpoint attributes are accessible through system Management.

FAL ASE:			**AR ASE**
CLASS:			**AR ENDPOINT**
CLASS ID:			**not used**
PARENT CLASS:			**TOP**
ATTRIBUTES:			
1	(m)	Attribute:	FAL Revision
2	(m)	Attribute:	Dedicated (TRUE, FALSE)
3	(m)	Attribute:	Cardinality (one-to-one, one-to-many)
4	(m)	Attribute:	Conveyance policy (Queued, Buffered)
5	(m)	Attribute:	Conveyance path (Unidirectional, Bi-directional)
6	(m)	Attribute:	Trigger policy (User-triggered, Network-scheduled)
7	(o)	Attribute:	Transfer Syntax
SERVICES:			
1	(o)	OpsService:	AR-Unconfirmed Send
2	(o)	OpsService:	AR-Confirmed Send
3	(o)	OpsService:	AR-Establish
4	(o)	OpsService:	AR-Abort

5.7.2.2 Attributes

FAL revision
This specifies the revision level of the FAL protocol used by this endpoint. The revision level is in the AR header of all FAL-PDUs transmitted.

Dedicated
This attribute specifies whether the endpoint is dedicated or not. When TRUE, the services of the AR ASE are accessed directly by the FAL User.

Cardinality
This attribute specifies cardinality of the AR described in 2.7.1.

Conveyance policy
This attribute specifies conveyance policy of the AR described in 2.7.1.

Conveyance path
This attribute specifies conveyance path of the AR described in 2.7.1.

Trigger policy
This attribute specifies trigger policy of the AR described in 2.7.1.

Transfer syntax
This optional attribute identifies the encoding rules to be used on the AR. When not present, the default FAL Transfer Syntax of this standard is used.

5.7.2.3 Services

All services defined for this class are optional. When an instance of the class is defined, at least one of these services should be selected.

AR-unconfirmed send
This optional service is used to send an unconfirmed service.

AR-confirmed send
This optional service is used to send a confirmed service.

AR-establish
This optional service is used to establish (open) an AR.

AR-abort
This optional service is used to abort (abruptly terminate) an AR.

5.7.3 Application relationship ASE service specification

5.7.3.1 Supported services

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

- AR-Unconfirmed Send
- AR-Confirmed Send
- AR-Establish
- AR-Abort

The services AR-Confirmed Send contain the FAL PDU body as part of the Result parameter in the response and confirmation primitives. The FAL PDU body may contain either the positive response or negative response returned by the FAL User transparently to the AR ASE. Therefore, these services have a single Result parameter instead of the separate Result(+) and Result(–) parameters that are commonly used to convey the positive and negative responses returned by the FAL User.

5.7.3.2 AR-Unconfirmed send service

5.7.3.2.1 Service overview

This service is used to send AR-Unconfirmed Send request APDUs for FAL APO ASEs. The AR-Unconfirmed Send service may be requested at either endpoint of a one-to-one bi-directional AR, at the server endpoint of a one-to-one unidirectional AR, or at the publisher endpoint of a one-to-many AR.

5.7.3.2.2 Service primitives

The service parameters for each primitive are shown in Table 20.

Table 20 – AR-Unconfirmed Send

Parameter name	Req	Ind
Argument		
AREP	M	M
Remote DL Address	M	M (=)
FAL Service Type	M	M (=)
FAL APDU Body	M	M (=)

Remote DL address
This parameter specifies the destination DLSAP address in the request and the source DLSAP address in the indication.

FAL service type
This parameter specifies the type of service being conveyed.

FAL APDU body
This parameter specifies the service-dependent body for the APDU.

5.7.3.2.3 Service procedure

The AR-Unconfirmed Send service is a service that operates through a queue.

The requesting FAL ASE submits an AR-Unconfirmed Send request primitive to its AR ASE. The AR ASE builds an AR-Unconfirmed Send request APDU.

The AR ASE queues the APDU for submission to the lower layer.

If the AREP is user-triggered, the AR ASE immediately requests the lower layer to transfer the APDU. If the AR is network-scheduled, the AR ASE requests the DL to transfer the data at the scheduled time. The data-link mapping indicates how the AR ASE coordinates its requests to transmit the data with the data-link layer.

Upon receipt of the AR-Unconfirmed Send request APDU, the receiving AR ASE delivers an AR-Unconfirmed Send indication primitive to the appropriate FAL ASE as indicated by the FAL Service Type Parameter.

5.7.3.3 AR-confirmed send service

5.7.3.3.1 Service overview

This service is used to send confirmed request and response APDUs for FAL APO ASEs. The AR-Confirmed Send service may be requested at Client and Peer endpoints of one-to-one bi-directional ARs.

5.7.3.3.2 Service primitives

The service parameters for each primitive are shown in Table 21.

Table 21 – AR-confirmed send

Parameter name	Req	Ind	Rsp	Cnf
Argument				
AREP	M	M		
Server DL-Address	M	M (=)		
FAL Service Type	M	M (=)		
FAL APDU Body	M	M (=)		
Result				
Client DL-Address			M	M (=)
FAL Service Type			M	M (=)
FAL APDU Body			M	M (=)
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. The method by which a response primitive is correlated with its corresponding preceding indication primitive is a local matter. See 1.2.				

Server DL-address
This parameter specifies the DL-address of the server.

FAL service type
This parameter specifies the type of service being conveyed.

FAL APDU body
This parameter specifies the service-dependent body for the APDU.

Result
This parameter indicates that the service request succeeded or failed.

Client DL-address
This parameter specifies the DL-address of the client.

5.7.3.3.3 Notional service procedure [INFORMATIVE]

NOTE The following notional explanation is a protocol matter and is outside the scope of this abstract service definition. It is included to indate one what in which these services could be provided by a real protocol.

The AR-Confirmed Send service is a service that operates through a queue.

The requesting FAL ASE submits an AR-Confirmed Send request primitive to its AR ASE. The AR ASE creates a transaction state machine to control the invocation of the service and assigns an InstanceID to that state machine.

The AR ASE builds an AR-Confirmed Send request APDU that includes the InstanceID and queues the APDU for submission to the lower layer, then the AR ASE immediately requests the lower layer to transfer the APDU.

Upon receipt of the AR-Confirmed Send request APDU, the receiving AR ASE creates a transaction state machine to manage the expected response, assigns an independent (second) InstanceID to that state machine, then delivers an AR-Confirmed Send indication primitive to the appropriate FAL ASE as indicated by the FAL Service Type Parameter, with the (second) InstanceID as an extra implementation parameter.

The responding FAL ASE submits a confirmed send response primitive to its AR ASE, identifying the transaction by the InstanceID presented as part of the response. The AR ASE builds a confirmed send response APDU containing the (first) InstanceID of the original requesting APDU.

The AR ASE queues the APDU for submission to the lower layer. In addition, the AR ASE immediately requests the lower layer to transfer the APDU.

Upon receipt of the confirmed send response APDU, the receiving AR ASE uses the (first) InstanceID contained in the response APDU to associate the response with the appropriate state machine and request. It then delivers an AR-Confirmed Send confirmation primitive to the requesting FAL ASE and cancels the associated transaction state machine.

If the timer expires before the sending AR ASE receives the response APDU, the AR ASE delivers an AR-Confirmed Send confirmation(–) primitive to the requesting FAL ASE and cancels the associated transaction state machine.

5.7.3.4 AR-establish service

5.7.3.4.1 Service overview

This confirmed service operates in a pair-wise manner between two AR endpoints to synchronize their contexts and activate them for the transfer of APDUs.

The endpoint context may be created during or prior to establishment through System Management. Once defined, the Set Attributes and Get Attributes services can be used to update and further coordinate endpoint contexts.

5.7.3.4.2 Service primitives

The service parameters for each primitive are shown in Table 28

Table 22 – AR-establish service

Parameter name	Req	Cnf
Argument		
AREP	M	
Remote DL-Address	M	
Result		
Error Info		C

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

AREP
This parameter specifies sufficient information to locally identify the AREP to be established.

Remote DL-address
This parameter identifies the remote DL-address.

Result
This parameter indicates whether the service request succeeded or failed.

5.7.3.4.3 Service procedure

5.7.3.4.3.1 AR establishment

The AR-Establish service causes local AREPs to be established independently.

Upon receipt of an AR-Establish request service primitive, the calling FAL issues an AR-Establish confirmation service primitive to the calling AP indicating whether or not it was able to establish the AREP. If one of the following cases is found, it was not able to establish the AREP:

a) the requested AREP is not specified within the FAL, or

b) resources are not available to establish the requested AREP, or

c) for AREPs supported by a data-link connection, the data-link layer was not able to establish the data-link connection.

5.7.3.5 AR-abort service

5.7.3.5.1 Service overview

The service is used by the FAL User to terminate an AR abruptly. It is always successful; the receiver of an Abort request always aborts the AR. This service may be used on any open AREP, regardless of whether the AR was pre-established or established dynamically using the establish service.

The AR-Abort service is used to instruct the AR ASE to abruptly terminate all activity on an AREP and place it in the closed state. Receipt of an AR-Abort request primitive causes the AR ASE to close the AREP context immediately. The immediate close of each endpoint context causes all outstanding service requests to be cleared. All subsequent service primitives and APDUs received by the AR ASE for the aborted AR are discarded except those of the AR-Establish service.

The AR ASE may also initiate the Abort service if it detects unrecoverable communication failures. In this case, the AR ASE delivers an AR-Abort indication primitive informing the user of the failure and closes the endpoint context.

5.7.3.5.2 Service primitives

The service parameters for each primitive are shown in Table 23.

Table 23 – AR-abort

Parameter name	Req	Ind
Argument		
AREP	M	M
Originator	M	M (=)
Reason Code	M	M (=)
Additional Detail	U	U (=)

Originator
This parameter identifies the originator of the AR-Abort. Possible values are “DLL”, “FAL”, or “FAL-USER”. The value DLL cannot be used in the request primitive.

Reason code
This parameter indicates the reason for the Abort. It may be supplied by either the provider or the user. One reason is defined: AR ASE error. Other reason code values may be supplied by the data-link layer or by the user.

Additional detail
This optional parameter specifies user data that accompanies the indication. When used, the value submitted in the request primitive is delivered unchanged in the indication primitive.

5.7.3.5.3 Service procedure

The Abort service is a service that operates through a queue.

If the user wishes to abort an AR, it submits an Abort request primitive to its FAL AR ASE. If an AR ASE detects an unrecoverable local failure or a communication failure, it delivers an abort indication primitive to the endpoint user.

The FAL AR ASE also transitions the AREP state to CLOSED.

6 ARs

6.1 General

Application Relationships provided are

- point-to-point user-triggered confirmed client/server AREP (PTC-AR);
- point-to-point user-triggered unconfirmed client/server AREP (PTU-AR);
- point-to-point network-scheduled unconfirmed publisher/subscriber AREP (PSU-AR);
- multipoint user-triggered unconfirmed publisher/subscriber AREP (MTU-AR);
- multipoint network-scheduled unconfirmed publisher/subscriber AREP (MSU-AR).

6.2 Point-to-point user-triggered confirmed client/server AREP (PTC-AR)

6.2.1 Overview

This class is defined to support the on-demand exchange of confirmed services between two application processes. Unconfirmed services are not supported by this type of AR. It uses connectionless-mode data-link services for the exchanges. The data-link layer transfers may be either acknowledged or unacknowledged. The behaviour of this class is described as follows. An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP and the AREP sending the request APDU queues it to its underlying layer for transfer at the next available opportunity.

The AREP receiving the request APDU from its underlying layer queues it for delivery to its AR ASE user in the order in which it was received.

The AREP receiving the request APDU accepts the corresponding response APDU from its AR ASE user and queues it to the underlying layer for transfer.

The AREP that issued the request APDU receives the response APDU from its underlying layer and queues it for delivery to its AR ASE user in the order in which it was received. It also stops its associated service response timer.

The characteristics of this AREP class are summarized as follows:

Roles	Client
	Server
	Peer
Cardinality	One-to-one
Conveyance paths	Bi-directional

Trigger policy	User-triggered
Conveyance policy	Queued

6.2.2 Formal model

FAL ASE:			**AR ASE**
CLASS:			**PTC-AR**
CLASS ID:			**not used**
PARENT CLASS:			**AR ENDPOINT**
ATTRIBUTES:			
1	(m)	Attribute:	Role (CLIENT, SERVER, PEER)
2	(m)	Attribute:	AREP State
3	(m)	Attribute:	Maximum Outstanding Requests Calling
4	(m)	Attribute:	Maximum Outstanding Requests Called
5	(m)	Attribute:	Transmit DL Mapping Reference
6	(m)	Attribute:	Receive DL Mapping Reference
SERVICES:			
1	(m)	OpsService:	Confirmed Send

6.2.3 Attributes

Role
This attribute specifies the role of the AREP. Possible values are

PEER (CLIENT/SERVER) Endpoints of this type may perform as either clients or servers, or as both simultaneously.

CLIENT Endpoints of this type issue confirmed service Request-APDUs to servers and receive confirmed service Response-APDUs.

SERVER Endpoints of this type receive confirmed service Request-APDUs from clients and issue confirmed service Response-APDUs to them.

AREP state
This attribute specifies the state of the AREP. The values for this attribute are specified in part 3 of this document.

Max outstanding requests calling
This conditional attribute indicates the maximum number of responses that the AREP may be expecting from the remote AREP.

Max outstanding requests called
This conditional attribute indicates the maximum number of responses that the AR endpoint may be expecting from its local user.

Transmit DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the transmit conveyance path for this AREP. DL mappings for the data-link layer are specified in part 3 of this document.

Receive DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the receive conveyance path for this AREP. DL mappings for the data-link layer are specified in part 3 of this document.

6.2.4 Services

Confirmed send
This optional service is used to send a confirmed service on an AR.

6.3 Point-to-point user-triggered unconfirmed client/server AREP (PTU-AR)

6.3.1 Overview

This class is defined to support the on-demand exchange of unconfirmed services between two application processes. It uses connectionless-mode data-link services for the exchanges. The data-link layer transfers may be either acknowledged or unacknowledged. The behaviour of this class is described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP. The AREP sending the request APDU queues it to its underlying layer for transfer at the next available opportunity.

The AREP receiving the request APDU from its underlying layer queues it for delivery to its AR ASE user in the order in which it was received.

The characteristics of this AREP class are summarized as follows:

Roles	Client
	Server
	Peer
Cardinality	One-to-one
Conveyance paths	Unidirectional
Trigger policy	User-triggered
Conveyance policy	Queued

6.3.2 Formal model

FAL ASE:			**AR ASE**
CLASS:			**Point-to-point user-Triggered Unconfirmed client/server AREP**
CLASS ID:			**not used**
PARENT CLASS:			**AR ENDPOINT**
ATTRIBUTES:			
1	(m)	Attribute:	Role (CLIENT, SERVER, PEER)
2	(m)	Attribute:	AREP State
3	(m)	Attribute:	Transmit DL Mapping Reference
4	(m)	Attribute:	Receive DL Mapping Reference
SERVICES:			
1	(m)	OpsService:	Unconfirmed Send

6.3.3 Attributes

Role

This attribute specifies the role of the AREP. Possible values are

PEER (CLIENT/SERVER) Endpoints of this type may perform as either clients or servers, or as both simultaneously. Endpoints of this type indicate whether they are to be the initiator of the AR establishment process.

CLIENT Endpoints of this type issue unconfirmed service Request-APDUs to servers.

SERVER Endpoints of this type receive confirmed and unconfirmed service Request-APDUs from clients.

AREP state
This attribute specifies the state of the AREP. The values for this attribute are specified in IEC61158-6-17.

Transmit DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the transmit conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

Receive DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the receive conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

6.3.4 Services

Unconfirmed send
This optional service is used to send an unconfirmed service on an AR.

6.4 Point-to-point network-scheduled unconfirmed publisher/subscriber AREP (PSU-AR)

6.4.1 Overview

This class is defined to support the “push” model for scheduled and buffered distribution of unconfirmed services to one application process.

The behaviour of this type of AR can be described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP for distribution. Sending AREP writes the APDU into the internal buffer, completely replacing the existing contents of the buffer. The AR ASE transfers the buffer contents at the next scheduled transfer opportunity.

If the AREP receives another APDU before the buffer contents are transmitted, the buffer contents will be replaced with the new APDU, and the previous APDU will be lost. When the buffer contents are transmitted, the AR ASE notifies the user of transmission.

At the receiving endpoint, the APDU is received from the network and is written immediately into the buffer, completely overwriting the existing contents of the buffer. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface. If the APDU has not been delivered before the next APDU arrives, it will be overwritten by the next APDU and lost.

An FAL user receiving the buffered transmission may request to receive the currently buffered APDU later.

The characteristics of this AREP class are summarized as follows:

Roles	Publisher
	Subscriber
Cardinality	One-to-one
Conveyance paths	Unidirectional
Trigger policy	Network-scheduled
Conveyance policy	Buffered

6.4.2 Formal model

FAL ASE:			**AR ASE**
CLASS:			**Point-to-point network-Scheduled Unconfirmed publisher/subscriber AREP**
CLASS ID:			**not used**
PARENT CLASS:			**AR ENDPOINT**
ATTRIBUTES:			
1	(m)	Attribute:	Role (PUBLISHER, SUBSCRIBER)
2	(m)	Attribute:	AREP State
3	(m)	Attribute:	Transmit DL Mapping Reference
4	(m)	Attribute:	Receive DL Mapping Reference
SERVICES:			
1	(o)	OpsService:	Unconfirmed Send

6.4.3 Attributes

Role
This attribute specifies the role of the AREP. Possible values are

PUBLISHER Endpoints of this type issue unconfirmed service Request-APDUs to subscribers.

SUBSCRIBER Endpoints of this type receive unconfirmed service Request-APDUs issued by the publisher.

AREP state
This attribute specifies the state of the AREP. The values for this attribute are specified in IEC61158-6-17.

Transmit DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the transmit conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

Receive DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the receive conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

6.4.4 Services

Unconfirmed send
This optional service is used to send an unconfirmed service on an AR.

6.5 Multipoint user-triggered unconfirmed publisher/subscriber AREP (MTU-AR)

6.5.1 Overview

This class is defined to support the on-demand exchange of unconfirmed services between two or more application processes. This class uses connectionless data-link services for the exchanges. The behaviour of this class is as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP. The AREP sending the request APDU queues it to its underlying layer for transfer at the next available opportunity.

The AREP receiving the request APDU from its underlying layer queues it for delivery to its AR ASE user in the order in which it was received.

The characteristics of this AREP class are summarized as follows:

Roles	Publisher
	Subscriber
Cardinality	One-to-many
Conveyance paths	Unidirectional
Trigger policy	User-triggered
Conveyance policy	Queued

6.5.2 Formal model

FAL ASE:			**AR ASE**
CLASS:			**Multipoint user-Triggered Unconfirmed publisher/subscriber AREP**
CLASS ID:			**not used**
PARENT CLASS:			**AR ENDPOINT**
ATTRIBUTES:			
1	(m)	Attribute:	Role (PUBLISHER, SUBSCRIBER)
2	(m)	Attribute:	AREP State
3	(m)	Attribute:	Transmit DL Mapping Reference
4	(m)	Attribute:	Receive DL Mapping Reference
SERVICES:			
1	(m)	OpsService:	Unconfirmed Send

6.5.3 Attributes

Role
This attribute specifies the role of the AREP. Possible values are

PUBLISHER Endpoints of this type issue unconfirmed service Request-APDUs to subscribers.

SUBSCRIBER Endpoints of this type receive unconfirmed service Request-APDUs issued by the publisher.

AREP state
This attribute specifies the state of the AREP. The values for this attribute are specified in IEC61158-6-17.

Transmit DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the transmit conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

Receive DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the receive conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

6.5.4 Services

Unconfirmed send
This optional service is used to send an unconfirmed service on an AR.

6.6 Multipoint network-scheduled unconfirmed publisher/subscriber AREP (MSU-AR)

6.6.1 Overview

This class is defined to support the “push” model for scheduled and buffered distribution of unconfirmed services to one or more application processes.

The behaviour of this type of AR can be described as follows.

An AR ASE user wishing to convey a request APDU submits it as an AR ASE Service Data Unit to its AREP for distribution. The sending AREP writes the APDU into the internal buffer, completely replacing the existing contents of the buffer.

The AREP transfers the buffer contents at the next scheduled transfer opportunity.

If the AREP receives another APDU before the buffer contents are transmitted, the buffer contents will be replaced with the new APDU, and the previous APDU will be lost. When the buffer contents are transmitted, the AR ASE notifies the user of transmission.

At the receiving endpoint, the APDU is received from the network and is written immediately into the buffer, completely overwriting the existing contents of the buffer. The endpoint notifies the user that the APDU has arrived and delivers it to the user according to the local user interface. If the APDU has not been delivered before the next APDU arrives, it will be overwritten by the next APDU and lost.

An FAL user receiving the buffered transmission may request to receive the currently buffered APDU later.

The characteristics of this AREP class are summarized as follows:

Roles	Publisher Subscriber
Cardinality	One-to-many
Conveyance paths	Unidirectional
Trigger policy	Network-scheduled
Conveyance policy	Buffered

6.6.2 Formal model

FAL ASE:			**AR ASE**
CLASS:			**Multipoint network-Scheduled Unconfirmed publisher/subscriber AREP**
CLASS ID:			**not used**
PARENT CLASS:			**AR ENDPOINT**
ATTRIBUTES:			
1	(m)	Attribute:	Role (PUBLISHER, SUBSCRIBER)
2	(m)	Attribute:	AREP State
3	(m)	Attribute:	Transmit DL Mapping Reference
4	(m)	Attribute:	Receive DL Mapping Reference
SERVICES:			
1	(o)	OpsService:	Unconfirmed Send

6.6.3 Attributes

Role
This attribute specifies the role of the AREP. Possible values are

PUBLISHER Endpoints of this type publish their data issuing unconfirmed service Request-APDUs.

SUBSCRIBER Endpoints of this type receive unconfirmed service Request-APDUs issued by publishers.

AREP state
This attribute specifies the state of the AREP. The values for this attribute are specified in IEC61158-6-17.

Transmit DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the transmit conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

Receive DL mapping reference
This attribute provides a reference to the underlying data-link layer mapping for the receive conveyance path for this AREP. DL mappings for the data-link layer are specified in IEC61158-6-17.

6.6.4 Services

Unconfirmed Send
This optional service is used to send an unconfirmed service on an AR.

7 Summary of FAL classes

This subclause contains a summary of the defined FAL Classes. The Class ID values have been assigned to be compatible with existing standards.

Table 30 provides a summary of the classes.

Table 24 – FAL class summary

FAL ASE	Class
Variable	Variable
	Variable list
Event	Notifier
Load region	Load region
Function invocation	Function invocation
Time	Time
Network management	Network manager
Application relationship	AREP
	PTC-AREP
	PTU-AREP
	PSU-AREP
	MTU-AREP
	MSU-AREP

8 Permitted FAL services by AREP role

Table 25 below defines the valid combinations of services and AR types (which service APDUs) and can be sent or received by AR with the specified type). “Unc” and “Cnf” columns indicate whether the service listed in the left-hand column is unconfirmed or confirmed respectively.

Table 25 – FAL services by AR type

	Unc	Cnf	Client		Server		Publisher		Subscriber	
FAL Services			**req**	**rcv**	**req**	**rcv**	**req**	**rcv**	**req**	**rcv**
Variable ASE										
Read		X	X			X				
Write		X	X			X				
Information Report	X				X		X			X
Event ASE										
Event Notification	X						X			X
Notification Recovery	X							X	X	
Load Region ASE										
Download		X	X			X				
Upload		X	X			X				
Function invocation ASE										
Start		X	X			X				
Stop		X	X			X				
Resume		X	X			X				
Time ASE										
Get Network Time		X								
Set Network Time	X						X			X
Tick Notification service	X						X			X
NW Management ASE										
Get Network Status		X								
Get Station Status		X								
NW Status Report	X						X			X
Station Status Report	X						X			X
AR ASE										
PTC-AR		X	X			X				
PTU-AR	X		X			X				
PSU-AR	X			X	X					
MTU-AR	X						X			X
MSU-AR	X						X			X

Bibliography

IEC 61158-6-17, *Industrial communication networks – Fieldbus specifications – Part 6-17: Application layer protocol specification – Type 17 elements*

IEC 61784-1 (Ed.2.0), *Industrial communication networks – Profiles – Part 1: Fieldbus profiles*

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
P.O. Box 131
CH-1211 Geneva 20
Switzerland

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