IEC 61158-5-26

Edition 2.0 2023-03

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 5-26: Application layer service definition – Type 26 elements

IEC 61158-5-26:2023-03(en)

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

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

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

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

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

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

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 300 terminological entries in English and French, with equivalent terms in 19 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC 61158-5-26

Edition 2.0 2023-03

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 5-26: Application layer service definition – Type 26 elements

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.70; 35.110

ISBN 978-2-8322-6581-9

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

FOREWORD 5
INTRODUCTION 7
1 Scope 8
1.1 General 8
1.2 Specifications 9
1.3 Conformance 9
2 Normative references 9
3 Terms, definitions, symbols, abbreviated terms and conventions 11
3.1 Terms and definitions from other ISO/IEC standards 11
3.1.1 Terms and definitions from ISO/IEC 7498-1 11
3.1.2 Terms and definitions from ISO/IEC 8822 11
3.1.3 Terms and definitions from ISO/IEC 9545 11
3.1.4 Terms and definitions from ISO/IEC 8824-1 12
3.2 Fieldbus application layer Type 26-specific definitions 12
3.3 Abbreviated terms and symbols 18
3.4 Conventions 20
3.4.1 Overview 20
3.4.2 General conventions 20
3.4.3 Conventions for class definitions 21
3.4.4 Conventions for service definitions 22
4 Concepts 23
5 Data type ASE 23
5.1 Overview 23
5.2 Formal definition of data type objects 24
5.2.1 Data type class definitions 24
5.2.2 Attributes 25
5.3 FAL defined data types 26
5.3.1 Fixed length types 26
5.3.2 String types 32
5.4 Data type ASE service specification 33
6 Communication model specification 33
6.1 General 33
6.2 Protocol stack for Type 26 fieldbus 33
6.3 Overview of Type 26 communication model 34
6.4 Cyclic data communication service with Common-memory 35
6.4.1 Overview 35
6.4.2 Common–memory: allocation to each node 36
6.4.3 Data sharing among nodes with the CM 37
6.4.4 CM data type 39
6.5 ASEs 39
6.5.1 Overview of Type 26 ASEs 39
6.5.2 Type 26 specific conventions for FAL service common parameters 40
6.5.3 Cyclic-data ASE 41
6.5.4 Message data ASE 47
6.5.5 Load measurement ASE 77
6.5.6 Network management ASE 83

6.5.7 General purpose command server ASE 97
6.5.8 AR ASE 100
6.5.9 FAL ASE summary 113
Bibliography 116

Figure 1 – Protocol stack for Type 26 fieldbus 34
Figure 2 – Unconfirmed Push-Publisher/Subscriber type interaction 35
Figure 3 – Unconfirmed/Confirmed Client/Server type interaction 35
Figure 4 – Common memory allocation 37
Figure 5 – Data sharing with the CM 38
Figure 6 – Node #01 for reception only 38
Figure 7 – Node #01 without the CM 39
Figure 8 − Data sharing among nodes with and without CM3 39
Figure 9 – The structure of ASEs for Type 26 FAL 40
Figure 10 – Virtual-address-space for Byte block 52
Figure 11 – Virtual-address-space for Word block 54
Figure 12 – AR ASE internal architecture 101
Figure 13 – Structure of IP address 111

Table 1 – Write service parameters 44
Table 2 – Send-CM service parameters 45
Table 3 – Read service parameters 45
Table 4 – Update memory service parameters 46
Table 5 – Get- buffer service parameters 47
Table 6 – Byte block read service parameters 52
Table 7 – Byte block write service parameters 53
Table 8 – Word block read service parameters 54
Table 9 – Word block write service parameters 55
Table 10 – Network parameter read service parameters 56
Table 11 – Extended network parameter read service parameters 57
Table 12 – Network parameter write service parameters 58
Table 13 – Extended network parameter write service parameters 59
Table 14 – Stop command service parameters 60
Table 15 – Operation command service parameters 60
Table 16 – profile read service parameters 61
Table 17 – Transparent message service parameters 63
Table 18 – Log data read service parameters 64
Table 19 – Log data items 64
Table 20 – Log data clear service parameters 69
Table 21 – Message return service parameters 70
Table 22 – Vendor specific message service parameters 71
Table 23 – Set remote node configuration parameter service parameters 72
Table 24 – Data elements and Node configuration parameters 72

Table 25 – Service parameters of Read remote participating node management information parameter service 73
Table 26 – Participating node management information parameters 74
Table 27 – Read remote node management information parameter service parameters 75
Table 28 – Node management information parameters 76
Table 29 – Read remote node setting information parameter service parameters 77
Table 30 – Node setting information parameters 77
Table 31 – Start TK-holding-time measurement service parameters 79
Table 32 – Terminate TK-holding-time measurement service parameters 80
Table 33 – Token-holding-time measurement result 81
Table 34 – Start GP_Comm sender log service parameters 82
Table 35 – Terminate GP_Comm sender log service parameters 82
Table 36 – GP_Comm sender log measurement result 83
Table 37 – Service parameters for Set configuration parameter 87
Table 38 – Configuration parameters 87
Table 39 – Read node management information parameter service parameters 88
Table 40 – Node management information parameters 88
Table 41 – Service parameters for Read participating node mgt. information parameter 89
Table 42 – Participating node management information parameters 90
Table 43 – Service parameters for Read network management information parameter 90
Table 44 – Network management information parameters 91
Table 45 – Service parameters for Read message sequence number management information 91
Table 46 – Read message sequence number management information parameters 92
Table 47 – Read node status service parameters 92
Table 48 – Read node status parameters 93
Table 49 – Upper layer operating condition matrix 93
Table 50 – Reset node service parameters 94
Table 51 – Set network address service parameters 94
Table 52 – Register service parameters 95
Table 53 – Event service parameters 95
Table 54 – Activate/Deactivate measurement service parameters 96
Table 55 – Get log data service parameters 97
Table 56 – Send command service parameters 100
Table 57 – CT send service parameters 104
Table 58 – MT send service parameters 106
Table 59 – CS send service parameters 107
Table 60 – Notify state change service parameters 108
Table 61 – Control measurement service parameters 109
Table 62 – DLSAP assignments 110
Table 63 – DLS Primitives and parameters 112
Table 64 – Lower layer T-profile and the required standards 113
Table 65 – Summary of FAL ASEs 113

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

INDUSTRIAL COMMUNICATION NETWORKS –
FIELDBUS SPECIFICATIONS –

Part 5-26: Application layer service definition –
Type 26 elements

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.

2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.

3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.

5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.

8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

Attention is drawn to the fact that the use of the associated protocol type is restricted by its intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a layer protocol type to be used with other layer protocols of the same type, or in other type combinations explicitly authorized by its intellectual-property-right holders.

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

IEC 61158-5-26 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

This second edition cancels and replaces the first edition published in 2019. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

a) expand Common-memory-area as a new Common-memory-area-3 (CM3);
b) add new services with expansion of Common-memory-area:
   - Extended-cyclic-data transfer service;
   - Extended-participation-request service;
   - Extended-network-parameter-read service;
   - Extended-network-parameter-write service.

The text of this International Standard is based on the following documents:

Draft	Report on voting
65C/1203/FDIS	65C/1244/RVD

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

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

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

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

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

INTRODUCTION

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

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

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

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-26: Application layer service definition – Type 26 elements

1 Scope

1.1 General

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

This part of IEC 61158 provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 26 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 document defines in an abstract way the externally visible service provided by the Type26 fieldbus application layer in terms of:

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

The purpose of this document is to define the services provided to:

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

This document specifies the structure and services of the Type 26 fieldbus application layer, in conformance with the OSI Basic Reference Model (see ISO/IEC 7498-1) and the OSI Application Layer Structure (see 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 document to provide access to the FAL to control certain aspects of its operation.

1.2 Specifications

The principal objective of this document 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, and the corresponding protocols standardized in subparts of IEC 61158-6.

This document can 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 document does not specify individual implementations or products, nor does it 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 26 application layer services as defined in this document.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

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

IEC 61158-1:2023, *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

IEC 61158-6-26:2023, *Industrial communication networks – Fieldbus specifications – Part 6-26: Application layer protocol specification – Type 26 elements*

IEC 61784-2-21:2023, *Industrial networks – Profiles – Part 2-21: Additional real-time fieldbus profiles based on ISO/IEC/IEEE 8802-3 – CPF 21*

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

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

ISO/IEC/IEEE 8802-3, *Telecommunications and exchange between information technology systems − Requirements for local and metropolitan area networks − Part 3: Standard for Ethernet*

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

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

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

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

ISO/IEC 60559, *Floating-Point arithmetic*

IETF RFC 768, J. Postel, *User Datagram Protocol*, August 1980, available at https://www.rfc-editor.org/info/rfc768 [viewed 2022-02-18]

IETF RFC 791, J. Postel, *Internet Protocol*, September 1981, available at https://www.rfc-editor.org/info/rfc791 [viewed 2022-02-18]

IETF RFC 792, J. Postel, *Internet Control Message Protocol*, September 1981, available at https://www.rfc-editor.org/info/rfc792 [viewed 2022-02-18]

IETF RFC 793, J. Postel, *Transmission Control Protocol*, September 1981, available at https://www.rfc-editor.org/info/rfc793 [viewed 2022-02-18]

IETF RFC 796, J. Postel, *Address mappings,* September 1981, available at https://www.rfc-editor.org/info/rfc796 [viewed 2022-02-18]

IETF RFC 826, D. Plummer, *An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware*, November 1982, available at https://www.rfc-editor.org/info/rfc826 [viewed 2022-02-18]

IETF RFC 894, C. Hornig, *A Standard for the Transmission of IP Datagrams over Ethernet*, April 1984, available at https://www.rfc-editor.org/info/rfc894 [viewed 2022-02-18]

IETF RFC 919, J.C. Mogul, *Broadcasting Internet Datagrams*, October 1984, available at https://www.rfc-editor.org/info/rfc919 [viewed 2022-02-18]

IETF RFC 922, J.C. Mogul, *Broadcasting Internet datagrams in the presence of subnets*, October 1984, available at https://www.rfc-editor.org/info/rfc922 [viewed 2022-02-18]

IETF RFC 950, J.C. Mogul and J. Postel, *Internet Standard Subnetting Procedure*, August 1985, available at https://www.rfc-editor.org/info/rfc950 [viewed 2022-02-18]

3 Terms, definitions, symbols, abbreviated terms and conventions

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

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

- IEC Electropedia: available at https://www.electropedia.org/
- ISO Online browsing platform: available at https://www.iso.org/obp

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 Terms and definitions from ISO/IEC 7498-1

For the purposes of this document, the following terms and definitions given in ISO/IEC 7498-1 apply.

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

3.1.2 Terms and definitions from ISO/IEC 8822

For the purposes of this document, the following terms and definitions given in ISO/IEC 8822 apply.

a) abstract syntax
b) presentation context

3.1.3 Terms and definitions from ISO/IEC 9545

For the purposes of this document, the following terms and definitions given in ISO/IEC 9545 apply.

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

3.1.4 Terms and definitions from ISO/IEC 8824-1

For the purposes of this document, the following terms and definitions given in ISO/IEC 8824-1 apply.

a) object identifier
b) type
c) value
d) simple type
e) structured type
f) tag
g) Boolean type
h) true
i) false
j) integer type
k) bitstring type
l) octetstring type

3.2 Fieldbus application layer Type 26-specific definitions

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

3.2.1
application
function or data structure for which data is consumed or produced

3.2.2
application objects
multiple object classes that manage and provide a run time exchange of messages across the network and within the network device

3.2.3
application process object
component of an application process that is identifiable and accessible through an FAL application relationship

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

3.2.4
application process object class
class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.5
application relationship
cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation

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

3.2.6
application relationship ASE
ASE used for establishing and terminating application relationship

3.2.7
application relationship endpoint
context and behavior of an application relationship as seen and maintained by one of the application processes involved in the application relationship

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

3.2.8
attribute
description of an externally visible characteristic or feature of an object

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

3.2.9
behavior
indication of how an object responds to particular events

3.2.10
channel
single physical or logical link of an input or output application object of a server to the process

3.2.11
class
set of objects, all of which represent the same kind of system component

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

3.2.12
class attributes
attribute that is shared by all objects within the same class

3.2.13
client
<object view> object which uses the services of another (server) object to perform a task

3.2.14
client
<communication view> initiator of a message to which a server reacts

3.2.15
common-memory
virtual memory accessible with logically unique address used for the cyclic-data transmission

Note 1 to entry: It is composed of the memory-area-1, -2 and -3 of the memory size 512 words, 8 192 words and 1 048 576 words respectively and is shared with the Type 26 nodes in a Type 26 fieldbus network.

3.2.16
connection
logical binding between application objects that are within the same or different devices

Note 1 to entry: Connections can be either point-to-point or multipoint.

3.2.17
consumer
node or sink that is receiving data from a producer

3.2.18
control commands
action invocations transferred from client to server to clear outputs, freeze inputs and/or synchronize outputs

3.2.19
cyclic
repetitive in a regular manner

3.2.20
cyclic-data frame
frame used for the Type 26 cyclic-data transmission

3.2.21
cyclic-data transmission
data transmission with which a Type 26 node sends out data on a pre-assigned area for the node on the common-memory, in broadcasting to all other Type 26 nodes every time the node obtains a token

3.2.22
dedicated AR
AR used directly by the FAL User

Note 1 to entry: On dedicated ARs, only the FAL Header and the user data are transferred.

3.2.23
device
physical hardware connected to the link

Note 1 to entry: A device can contain one or more than one node.

3.2.24
device profile
collection of device dependent information and functionality providing consistency between similar devices of the same device type

3.2.25
end node
producing or consuming node

3.2.26
endpoint
one of the communicating entities involved in a connection

3.2.27
error
discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.2.28
event
instance of a change of conditions

3.2.29
FA-link protocol
communication protocol for Type 26 fieldbus

Note 1 to entry: It defines the token management, the cyclic data transmission and the message data transmission, with which each Type 26 node performs data transmission on an equal basis.

3.2.30
FAL subnet
subnetworks composed of one or more data link segments, identified by a subset of the network address

3.2.31
frame
synonym for DLPDU

3.2.32
invocation
act of using a service or other resource of an application process

Note 1 to entry: Each invocation represents a separate thread of control that can 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.

3.2.33
index
address of an object within an application process

3.2.34
instance
actual physical occurrence of an object within a class that identifies one of many objects within the same object class

EXAMPLE California is an instance of the object class state.

Note 1 to entry: The terms object, instance, and object instance are used to refer to a specific instance.

3.2.35
instance attributes
attribute that is unique to an object instance and not shared by the object class

3.2.36
instantiated
object that has been created in a device

3.2.37
logical device
FAL class that abstracts a software component or a firmware component as an autonomous self-contained facility of an automation device

3.2.38
management information
network-accessible information that supports managing the operation of the fieldbus system, including the application layer

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

3.2.39
message-data frame
frame used for the Type 26 message-data transmission

3.2.40
message-data transmission
sporadic data transmission with which a Type 26 node sends out the message data of request, response, information report, notification upon event happened at the node and so forth, to other Type 26 nodes

3.2.41
method
synonym for an operational service which is provided by the server ASE and invoked by a client

3.2.42
module
hardware or logical component of a physical device

3.2.43
multipoint connection
connection from one node to many

Note 1 to entry: Multipoint connections allow messages from a single producer to be received by many consumer nodes.

3.2.44
network
set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.2.45
network load measurement function
provision for monitoring and measurement of the communication load over a Type 26 fieldbus with a Type 26 node

3.2.46
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 behavior

3.2.47
participation-request-frame
frame used for a Type 26 node notifying of its participation-request in a Type 26 fieldbus network

3.2.48
peer
role of an AR endpoint in which it is capable of acting as both client and server

3.2.49
physical device
automation or other network device

3.2.50
point-to-point connection
connection that exists between exactly two application objects

3.2.51
property
general term for descriptive information about an object

3.2.52
protocol type
identification of communication protocol

3.2.53
publisher
role of an AR endpoint that transmits APDUs onto the fieldbus for consumption by one or more subscribers

Note 1 to entry: A publisher cannot be aware of the identity or the number of subscribers and it can publish its APDUs using a dedicated AR.

3.2.54
pull publisher
type of publisher that publishes an object in response to a request received from its pull publishing manager

3.2.55
pull publishing manager
type of publishing manager that requests that a specified object be published in a corresponding response APDU

3.2.56
push publisher
type of publisher that publishes an object in an unconfirmed service request APDU

3.2.57
push publishing manager
type of publishing manager that requests that a specified object be published using an unconfirmed service

3.2.58
push subscriber
type of subscriber that recognizes received unconfirmed service request APDUs as published object data

3.2.59
resource
processing or information capability of a subsystem

3.2.60
sequence number
consecutive number used to determine whether a receiving message-data from other nodes has been previously received and is redundant

Note 1 to entry: The sequence number of a node is set to 0 (zero) at the time of the network-startup, and it is incremented by 1 (one) every time a message-data transmission is completed. The sequence number of the node is transmitted to other nodes by the message-data transmission.

3.2.61
server
<communication view> role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

3.2.62
server
<object view> object which provides services to another (client) object

3.2.63
service
operation or function than an object and/or object class performs upon request from another object and/or object class

3.2.64
subscriber
role of an AREP in which it receives APDUs produced by a publisher

3.2.65
token
right to send out data on the transmission medium, which is passed successively from node to node for Type 26 fieldbus network

3.2.66
token-frame
frame used for hand-over the token between Type 26 nodes

3.2.67
transaction code
code used to discriminate and identify each data transmission frame, and which represents the transaction of a receiving frame

3.2.68
trigger-frame
frame used in the network-startup state for a Type 26 node notifying of its entrance into a new participation-request over a Type 26 fieldbus network

3.2.69
valid-linking state
state in which a token is consecutively circulated among nodes over Type 26 fieldbus network

3.2.70
version of sequence number
version number of the sequence number in a receiving message-data from other nodes, used to determine whether the receiving message-data is valid and is not overdue

Note 1 to entry: The value is set at the time when a Type 26 node completes to participate in a Type 26 fieldbus network, and is maintained until the node drops out of the Type 26 fieldbus network.

3.2.71
virtual-address-space
virtual memory space with pseudo logical address, used for the message-transmission

3.3 Abbreviated terms and symbols

AE	Application entity
AL	Application layer
ALME	Application layer management entity
ALP	Application layer protocol
ALS	Application layer service
AP	Application process
APO	Application process object
APDU	Application protocol data unit
AR	Application relationship
AREP	Application relationship endpoint

ASCII	American Standard Code for Information Interchange
ASE	Application service element
BCT	Broadcast transmission
C_AD1	Data-head-address on common-memory-area-1
C_AD2	Data-head-address on common-memory-area-2
C_AD3	Data-head-address on common-memory-area-3
CIM	Computer integrated manufacturing
CM	Common-memory
CM1	Common-memory-area-1
CM2	Common-memory-area-2
CM3	Common-memory-area-3
Cnf	Confirmation
C_SZ1	Data-size on common-memory-area-1
C_SZ2	Data-size on common-memory-area-2
C_SZ3	Data-size on common-memory-area-3
DA	Destination-address
DL-	(as a prefix) data-link-
DLC	Data-link connection
DLCEP	Data-link connection endpoint
DLL	Data-link layer
DLS	Data-link service
DLSAP	Data-link service access point
DLSDU	DL-service-data-unit
DNA	Destination-node-number
DNS	Domain name service
FAL	Fieldbus application layer
HMI	Human-machine interface
H_TYPE	Header-type
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
IP	Internet protocol
ISO	International Organization for Standardization
LME	Layer management entity
LSB	Least Significant Bit
M_ADD	Message-offset-address in the virtual-address-space
MAC	Media Access Control
MAJ_VER	FA-link protocol version: major-version
M_CTL	Message-control
MFT	Allowable-minimum-frame-interval-time
MIN_VER	FA-link protocol version: minor-version
MSB	Most Significant Bit
MSN	Manufacturer's model name of a Type 26 node

M_SZ	Message-data-size in the virtual-address-space
M_RLT	Message result
NDN	Node name of Type 26 node
OSI	Open Systems Interconnect
PDU	Protocol data unit
PhL	Ph-layer
PTP	Point-to-Point
P_TYPE	Protocol-type
RCT	Allowable-refresh-cycle-time
Req	Request
RMT	Refresh-cycle-measurement-time
Rsp	Response
RT	Real Time
SA	Source-address
SAP	Service access point
SDU	Service data unit
SEQ	Sequence-number
SNA	Source-node-number
TCD	Transaction code
TCP	Transmission Control Protocol
TW	Token-watchdog
UDP	User Datagram protocol
ULS	Upper layer status
VDN	Vendor code of the node
V_SEQ	Version of sequence number

3.4 Conventions

3.4.1 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 ASE services specified in 6.5 of this document. The service specification defines the services that are provided by the ASE.

3.4.2 General conventions

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

3.4.3 Conventions for class definitions

Class definitions are described using templates. Each template consists of a list of attributes and services 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

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

b) The "CLASS:" entry is the name of the class being specified. All objects defined using this template are an instance of this class. The class may be specified by this document, or by a user of this document.

c) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that provides 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 document to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2 048 are allocated for identifying user defined classes.

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

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

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

   1) 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.

   2) 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.

   3) 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).

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

   1) An (m) in column 2 indicates that the service is mandatory for the class, while an (o) indicates that it is optional. A (c) in this column indicates that the service is conditional. When all services defined for a class are defined as optional, at least one has to be selected when an instance of the class is defined.

   2) The label "OpsService" designates an operational service (1).

   3) The label "MgtService" designates a management service (2).

   4) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting within the list of services is used to specify services conditional on a constraint statement.

3.4.4 Conventions for service definitions

3.4.4.1 General

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

3.4.4.2 Service parameters

Service primitives, used to represent service user/service provider interactions (see ISO/IEC 10731), convey parameters that indicate information available in the user/provider interaction. In any particular interface, not all parameters need be explicitly stated.

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

a) parameter name,

b) request primitive (transmitted from the sender),

c) indication primitive (transmitted to the receiver),

d) response primitive (transmitted from the receiver), and

e) confirm primitive (transmitted to the sender).

One parameter (or part 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 and parameter direction specified in the column:

M parameter is mandatory for the primitive

U parameter is a User option, and can be provided or not depending on the 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

1) 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, and

2) 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.

3.4.4.3 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 FAL.

4 Concepts

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 three parts: the class definitions, the services and the protocol specification. The first two are contained in this document. The protocol specification for each of the ASEs is defined in IEC 61158-6-26.

The fundamental concepts and templates used to describe the application layer service in this document follow IEC 61158-1, Clause 1.

The FAL defined in this document identifies a peer-based controller communications network supporting a Common-memory model in 6.4 as well as client/server models.

5 Data type ASE

5.1 Overview

Fieldbus data types specify the machine independent syntax for application data conveyed by FAL services. The FAL supports the definition and transfer of both basic and constructed data types. Basic types are atomic types. Constructed types are types composed of basic types and other constructed types.

IEC 61158-1, 10.1 provides an overview of the data type ASE and the relationship between data types, and is incorporated herein by reference.

Most basic types are defined from a set of ISO/IEC 8824 types (simple types):

- Basic types have a constant length;
- Two variations are defined, one for defining data types whose length is an integral number of octets, and one for defining data types whose length is bit;
- Some ISO/IEC 8824 types have been extended for fieldbus specific use (specific types). They are defined as simple class subtypes.

Constructed types are of strings, arrays and structures:

- A string shall be composed of an ordered set, variable in number, of homogeneously typed fixed-length elements;
- An array shall be composed of an ordered set of homogeneously typed elements. There are no restrictions on the data type of array elements placed, but each element shall be of the same type;
- A structure shall be made of an ordered set of heterogeneously typed elements called fields. Like arrays, this document does not restrict the data type of fields. However, the fields within a structure do not have to be of the same type.

Data types are defined as instances of the data type class. Defining new types shall be accomplished by providing a new numeric id and supplying values for the attributes defined for the data type class.

User data is transferred between applications by the FAL protocol. All encoding and decoding is performed by the FAL user. The rules for encoding user data in FAL protocol data units are data type dependent. These rules are defined in IEC 61158-6-26.

5.2 Formal definition of data type objects

5.2.1 Data type class definitions

The data type class specifies the root of the data type class tree. Its parent class "top" indicates the top of the FAL class tree. The general form of the template to describe the class definitions is shown below:

FAL ASE:			**DATA TYPE ASE**
CLASS:			**DATA TYPE**
CLASS ID:			5 (FIXED LENGTH & STRING), 6 (STRUCTURE), 12 (ARRAY)
PARENT CLASS:			TOP
ATTRIBUTES:			
1	(o)	Key Attribute:	Data Type Numeric Identifier
2	(o)	Key Attribute:	Data Type Name
3	(m)	Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
4	(c)	Constraint:	Format = FIXED LENGTH | STRING
4.1	(m)	Attribute:	Octet Length
5	(c)	Constraint:	Format = STRUCTURE
5.1	(m)	Attribute:	Number of Fields
5.2	(m)	Attribute:	List of Fields
5.2.1	(o)	Attribute:	Field Name
5.2.2	(m)	Attribute:	Field Data Type
6	(c)	Constraint:	Format = ARRAY

6.1	(m)	Attribute:	Number of Array Elements
6.2	(m)	Attribute:	Array Element Data Type

5.2.2 Attributes

Data type numeric identifier
This optional attribute identifies the numeric identifier of the related data type.

Data type name
This optional attribute identifies the name of the related data type.

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

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

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

List of fields
This conditional attribute is an ordered list of fields contained in the structure. Each field is specified by its number and its type. Fields shall be numbered sequentially from 0 (zero) in the order in which they occur. Partial access to fields within a structure is supported by identifying the field by number. This attribute shall be present when the value of the format attribute is "STRUCTURE".

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

Field data type
This conditional attribute specifies the data type of the field. It shall be present when the value of the format attribute is "STRUCTURE". This attribute can itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the embedded-data-type description shown below shall be used.

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

Array element data type
This conditional attribute specifies the data type for the elements of an array. All elements of the array shall have the same data type. It shall be present when the value of the format attribute is "ARRAY". This attribute can itself specify a constructed data type either by referencing a constructed data type definition by its numeric id, or by embedding a constructed data type definition here. When embedding a description, the embedded-data- type description shown below shall be used.

Embedded data type description
This attribute is used to recursively define embedded data types within a structure or array. The template below defines its contents. The attributes shown in the template are defined above in the data type class, except for the embedded-data-type attribute, which is a recursive reference to this attribute. It is used to define nested elements.

ATTRIBUTES:

1	(m)	Attribute:	Format (FIXED LENGTH, STRING, STRUCTURE, ARRAY)
2	(c)	Constraint:	Format = FIXED LENGTH | STRING
2.1	(m)	Attribute:	Data Type Numeric ID value
2.2	(m)	Attribute:	Octet Length
3	(c)	Constraint:	Format = STRUCTURE
3.1	(m)	Attribute:	Number of Fields
3.2	(m)	Attribute:	List of Fields
3.2.1	(m)	Attribute:	Embedded Data Type Description
4	(c)	Constraint:	Format = ARRAY
4.1	(m)	Attribute:	Number of Array Elements
4.2	(m)	Attribute:	Embedded Data Type Description

5.3 FAL defined data types

5.3.1 Fixed length types

5.3.1.1 Boolean types

5.3.1.1.1 Boolean

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	1
2	Data Type Name	=	Boolean
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

This data type expresses a Boolean data type with the values TRUE (non zero) and FALSE (zero).

5.3.1.1.2 BOOL

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

5.3.1.2 BitString types

5.3.1.2.1 BitString8

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	22
2	Data Type Name	=	BitString8
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	1

This type contains 1 element of type BitString.

5.3.1.2.2 BitString16

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	23
2	Data Type Name	=	BitString16
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	2

5.3.1.2.3 WORD

This IEC 61131-3 type is the same as BitString16.

5.3.1.2.4 BitString32

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	24
2	Data Type Name	=	BitString32
3	Format	=	FIXED LENGTH
5.1	Octet Length	=	4

5.3.1.2.5 DWORD

This IEC 61131-3 type is the same as BitString32.

5.3.1.3 Numeric types

5.3.1.3.1 Floating Point types

5.3.1.3.1.1 Float32

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	8
2	Data Type Name	=	Float32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

This type has a length of four octets. The format for float32 is that defined by ISO/IEC 60559 as single precision.

5.3.1.3.1.2 REAL

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

5.3.1.3.1.3 Float64

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

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

This type has a length of eight octets. The format for float64 is that defined by ISO/IEC 60559 as double precision.

5.3.1.3.2 Integer types

5.3.1.3.2.1 Integer8

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	2
2	Data Type Name	=	Integer8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

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

5.3.1.3.2.2 SINT

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

5.3.1.3.2.3 char

This data type is the same as Integer8.

5.3.1.3.2.4 Integer16

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	3
2	Data Type Name	=	Integer16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

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

5.3.1.3.2.5 INT

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

5.3.1.3.2.6 Integer32

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

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

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

5.3.1.3.2.7 DINT

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

5.3.1.3.2.8 Integer64

FAL ASE:		**DATA TYPE ASE**	
CLASS:		**DATA TYPE**	
CLASS ID:		5	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1	Data Type Numeric Identifier	=	55
2	Data Type Name	=	Integer64
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	8

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

5.3.1.3.2.9 LINT

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

5.3.1.3.3 Unsigned types

5.3.1.3.3.1 Unsigned8

FAL ASE:		**DATA TYPE ASE**	
CLASS:		**DATA TYPE**	
CLASS ID:		5	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1	Data Type Numeric Identifier	=	5
2	Data Type Name	=	Unsigned8
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	1

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

5.3.1.3.3.2 USINT

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

5.3.1.3.3.3 Unsigned16

FAL ASE:		**DATA TYPE ASE**	
CLASS:		**DATA TYPE**	
CLASS ID:		5	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1	Data Type Numeric Identifier	=	6
2	Data Type Name	=	Unsigned16
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	2

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

5.3.1.3.3.4 UINT

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

5.3.1.3.3.5 Unsigned32

FAL ASE:		**DATA TYPE ASE**	
CLASS:		**DATA TYPE**	
CLASS ID:		5	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1	Data Type Numeric Identifier	=	7
2	Data Type Name	=	Unsigned32
3	Format	=	FIXED LENGTH
4.1	Octet Length	=	4

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

5.3.1.3.3.6 UDINT

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

5.3.1.3.3.7 Unsigned64

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

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

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

5.3.1.3.3.8 ULINT

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

5.3.2 String types

5.3.2.1 BitString

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

1	Data Type Numeric Identifier	=	14
2	Data Type Name	=	BitString
3	Format	=	STRING
5.1	Octet Length	=	1 to n

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

5.3.2.2 OctetString

FAL ASE:	**DATA TYPE ASE**
CLASS:	**DATA TYPE**
CLASS ID:	5
PARENT CLASS:	TOP

ATTRIBUTES:

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

An OctetString is an ordered sequence of octets, numbered from 1 to n. Octet 1 of the sequence is referred to as the first octet. IEC 61158-6-26 defines the order of transmission.

5.3.2.3 VisibleString

FAL ASE:		**DATA TYPE ASE**	
CLASS:		**DATA TYPE**	
CLASS ID:		5	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1	Data Type Numeric Identifier	=	9
2	Data Type Name	=	VisibleString
3	Format	=	STRING
4.1	Octet Length	=	1 to n

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

5.4 Data type ASE service specification

There are no operational services defined for the type object.

6 Communication model specification

6.1 General

This document specifies the application layer service of Type 26 fieldbus essential for the ISO/IEC/IEEE 8802-3 based FA Link network, which is incorporated as one of the communication networks for the Real-Time Ethernet (RTE) defined in IEC 61784-2-21.

The Type 26 fieldbus meets the industrial automation market objective of providing predictable time deterministic and reliable time-critical data transfer and means, which allow co-existence with non-time-critical data transfer over the ISO/IEC/IEEE 8802-3 series communications medium, for support of cooperation and synchronization between automation processes on field devices in a real-time application system. 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.

This document specifies the Application layer service part of the protocol set of the communication profile and/or of one or more communication profiles related to the common family of CPF 21.

6.2 Protocol stack for Type 26 fieldbus

The application layer exploits the services of the data-link layer immediately below, in terms of the "three-layer" Fieldbus Reference Model which is based in part on the OSI Basic Reference Model. The OSI reference model provides a layered approach to communications standards, whereby the layers can be developed and modified independently. IEC 61158 specifies functionality from top to bottom of a full OSI stack and, potentially, some functions for the users of the stack.

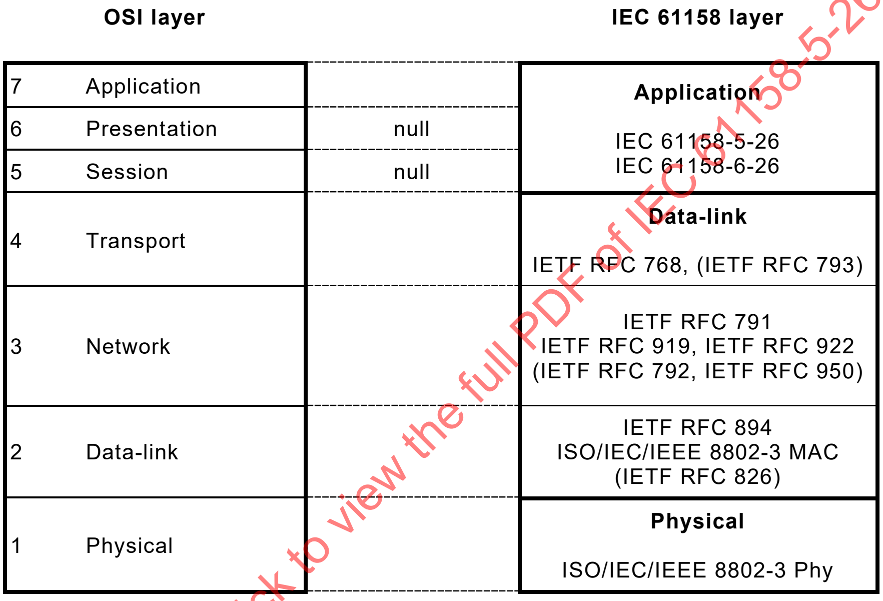
Figure 1 shows the protocol stack for a Type 26 fieldbus, the OSI layers and the equivalent layers in the IEC 61158 basic fieldbus reference model.

Functions of the intermediate OSI layers, layers 3 through 6, are consolidated into the IEC 61158 data-link layer for a Type 26 fieldbus as the lower layer of the FA Link protocol. The data-link layer is that of a general-purpose Ethernet with UDP (see IETF RFC 768), TCP (see IETF RFC 793), IP (see IETF RFC 791) and Ethernet layer (see IETF RFC 894, ISO/IEC/IEEE 8802-3) as the main components. Type 26 fieldbus utilizes that of the services provided with a general-purpose Ethernet protocol suite.

The data-link layer for Type 26 fieldbus shall specify a broadcast communication function as the mandatory fundamental function of the lower layer, and any other main function is not specified as mandatory for the lower layer protocol.

The application layer for Type 26 fieldbus is the main component of the FA Link protocol stack and is specified as the upper layer of the FA Link protocol. The application layer is mapped directly on the data-link layer.

Equivalent function of the OSI layer 5 and 6 are not used for a Type 26 fieldbus. Features common to users of the fieldbus application layer are provided by the IEC 61158 application layer to simplify user operation. This document defines the service elements and the services, and the protocol specification is defined in IEC 61158-6-26.



OSI layer			IEC 61158 layer
7	Application		**Application** IEC 61158-5-26 IEC 61158-6-26
6	Presentation	null	
5	Session	null	
4	Transport		**Data-link** IETF RFC 768, (IETF RFC 793)
3	Network		IETF RFC 791 IETF RFC 919, IETF RFC 922 (IETF RFC 792, IETF RFC 950)
2	Data-link		IETF RFC 894 ISO/IEC/IEEE 8802-3 MAC (IETF RFC 826)
1	Physical		**Physical** ISO/IEC/IEEE 8802-3 Phy

Figure 1 – Protocol stack for Type 26 fieldbus

6.3 Overview of Type 26 communication model

Two types of communication service models are used in FAL Type 26; i.e. the cyclic data- and the message data-communication service models as follows:

a) Cyclic data communication service model:

Communication service model using the n: n-type distributed-shared-virtual memory with a common pseudo logical address among the Type 26 nodes over a Type 26 fieldbus network, is shown in Figure 2.

This communication service is performed periodically with the Unconfirmed Push-Publisher/Subscriber type communications on a buffered AREP;

b) Message data communication service model:

Communication service model using the Unconfirmed/Confirmed Client/Server communications with Push type on queued AREPs is shown in Figure 3.

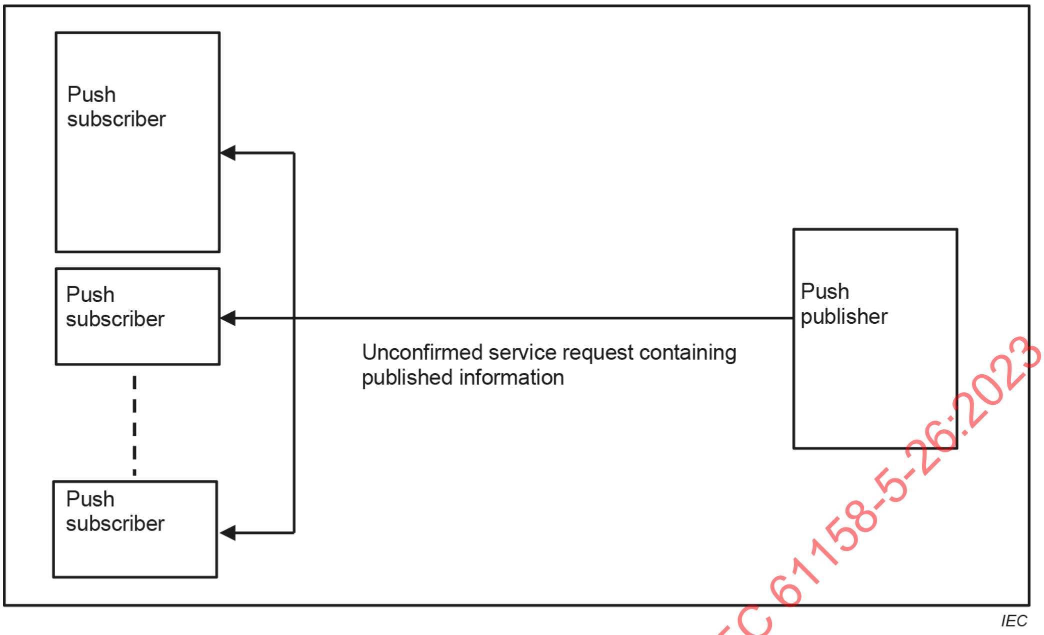

Push
subscriber
Push
subscriber
Push
subscriber
Push
publisher
Unconfirmed service request containing published information
IEC

Figure 2 – Unconfirmed Push-Publisher/Subscriber type interaction

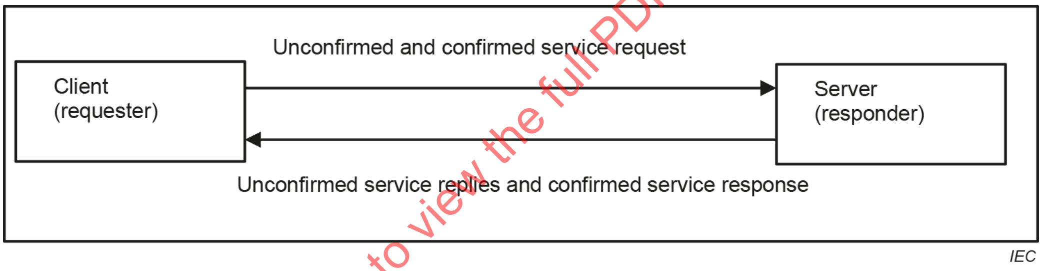

Unconfirmed and confirmed service request
Client
(requester)
Server
(responder)
Unconfirmed service replies and confirmed service response
IEC

Figure 3 – Unconfirmed/Confirmed Client/Server type interaction

6.4 Cyclic data communication service with Common-memory

6.4.1 Overview

The FAL defined in this document identifies a peer based controller communications network supporting a Common-memory model.

The Type 26 fieldbus is applicable to a controller level network for efficiently executing the control and state data exchange between various kinds of field devices such as PLCs (Programmable Logic Controllers), RCs (Robot Controllers), NCs (Numerical Controllers), FA (Factory Automation) controllers, DCSs (Distributed Control systems), PCs (Personal Computers) for the control purpose and the human-machine interfaces and so forth, in industrial automation control systems in which a time-criticality for their data exchanges is required.

The communications between these field devices also requires simplicity in actual application programming as well as the time-critical transfer of the control and state data required among these field devices.

The Common-memory (CM) provides a function to share the control and state data seamlessly across an industrial control network system.

6.4.2 Common–memory: allocation to each node

The CM is a virtual common memory accessible with logically unique address across the Type 26 nodes in a Type 26 fieldbus network, and the APs running over each Type 26 node can handle the CM as a global common memory shared among the Type 26 nodes over the Type 26 fieldbus network.

Cyclic-data transmission is used for refreshing and equalizing the contents of the CM among the Type 26 nodes by means of a cyclic broadcasting performed at a constant rate after a Type 26 node writes a data peace on a pre-allocated memory area of the CM for the Type 26 node.

Figure 4 shows the CM which is composed of three memory areas; i.e. the common-memory-area-1 (CM1), -2 (CM2) and -3 (CM3) with the memory size of 512 words, 8 192 words and 1 048 576 words respectively. The total memory size of the CM is up to 1 057 280 words. The CM3 is the expansion of CM1 and CM2.

The memory area allocated to each node as the output/write area on the CM1, CM2 and CM3 is dependent on the implementation.

Any memory size in word boundary is allocable to each node from 0 (zero) to the maximum space size of the CM1 and/or CM2, and as of the CM3 from 0 (zero) to maximum 32 768 words is allocable. The memory space allocation is designated with the memory size and the head-address. The memory space allocated to each node shall not overlap with each other. And the allocated memory space is identical to the cyclic-data transmission area by the node.

In case of total zero memory space allocated to a node, it means no cyclic-data transmission to be carried out by the node. The provision with the CM at each node is dependent on the implementation of each node, and it is applicable to a mixture system of the nodes with and without the CM.

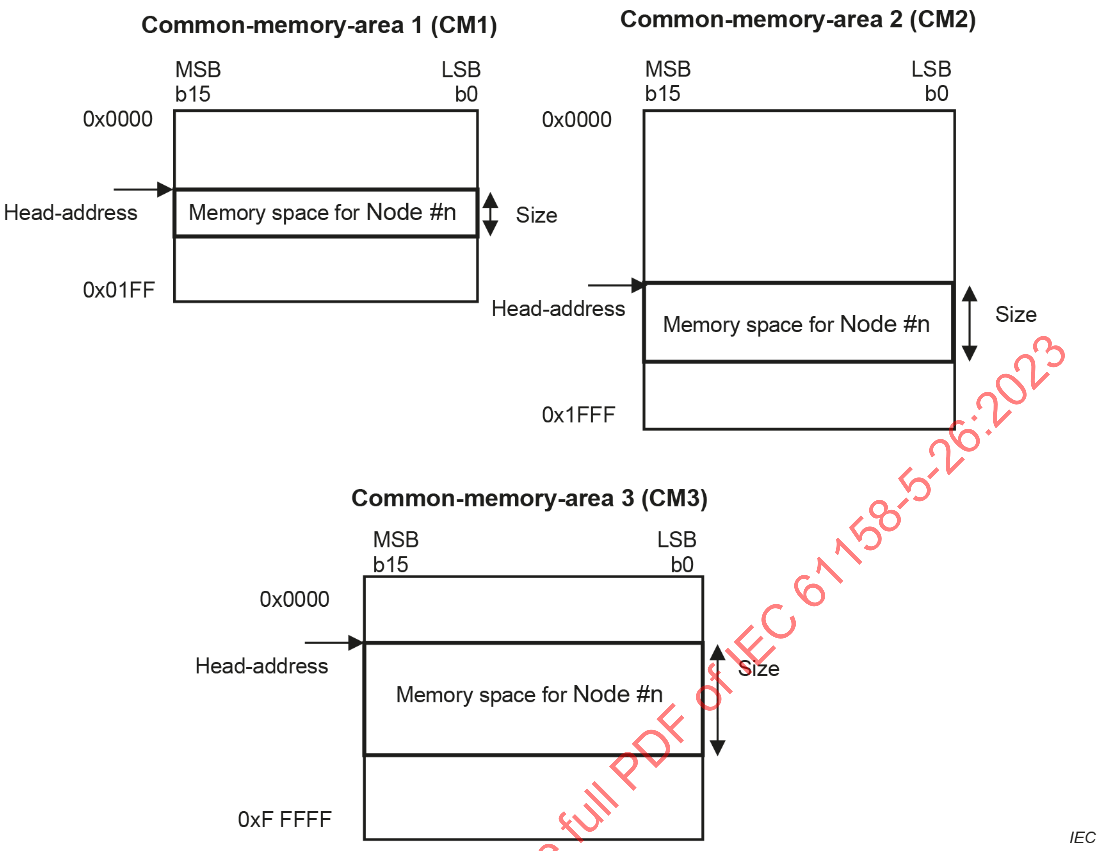


Figure 4 – Common memory allocation

6.4.3 Data sharing among nodes with the CM

The CM, by means of the cyclic-data transmission performed at a constant rate, provides the data distribution/refreshing function to all Type 26 nodes in temporal and spatial coherency, so that each node can share the same data with each other and can make good use of the data; i.e. a memory space allocated to each node is a container for application data in general use and provides flexibility to apply in a variety of industrial application processes.

In operational view of the CM, the memory area in the CM allocated to a certain node is used as the transmission area of the node by a cyclic-data transmission as a sender or a Push-publisher. On the other hand, as receiving nodes of the cyclic-data transmission or Push-subscribers, the memory area of the sender node is used as a receiving area for the data from the sender node.

Figure 5 shows the data sharing with the CM.

The memory area or the transmission area by a cyclic-data transmission, allocated on the CM, shall not be overlapped with each other.

It is available for a node to use the CM only as the receiving area for other nodes. Figure 6 shows an example in which the node #01 is of reception only.

In the case of a node without the CM, all the receiving data by a cyclic-data transmission shall be discarded. Figure 7 shows an example in which the node #01 is not the recipient or the Push-subscriber.

The CM3 is the expansion of the CM1 and CM2, and the Type 26 nodes with CM3 are available for the data sharing on the CM3 among those nodes, and only the nodes with CM3 can send out, receive the specific cyclic data frames or extended-cyclic-data frames to share the data on the CM3 among those nodes. When a node without CM3 receives any extended-cyclic-data frames, the extended-cyclic-data frame is discarded. Figure 8 shows the data sharing among the node #02 without CM3 and the node #1, #05 with CM3 as an example.

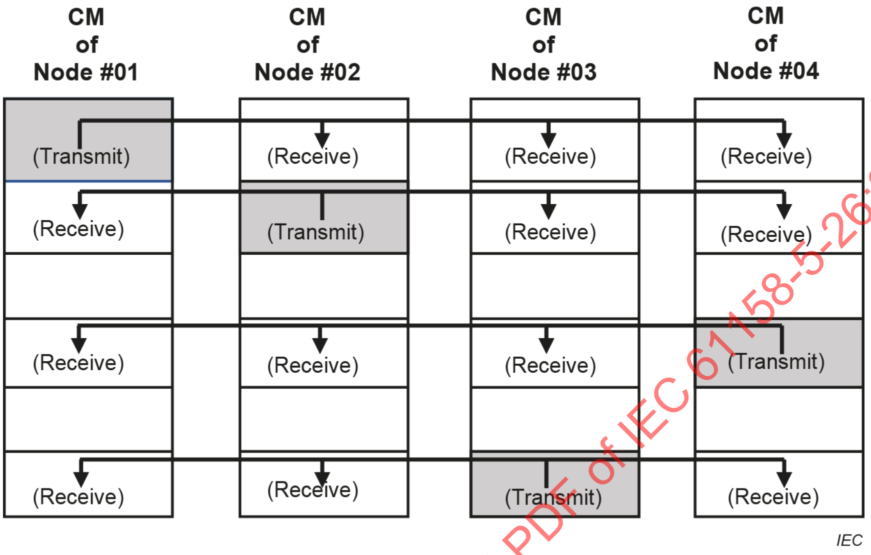


Figure 5 – Data sharing with the CM

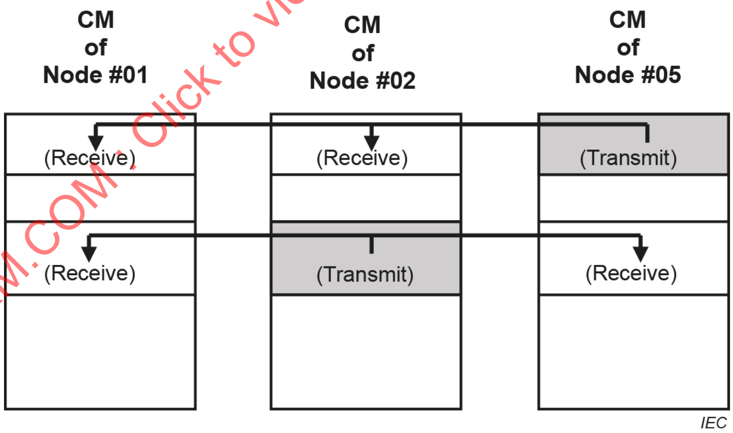


Figure 6 – Node #01 for reception only

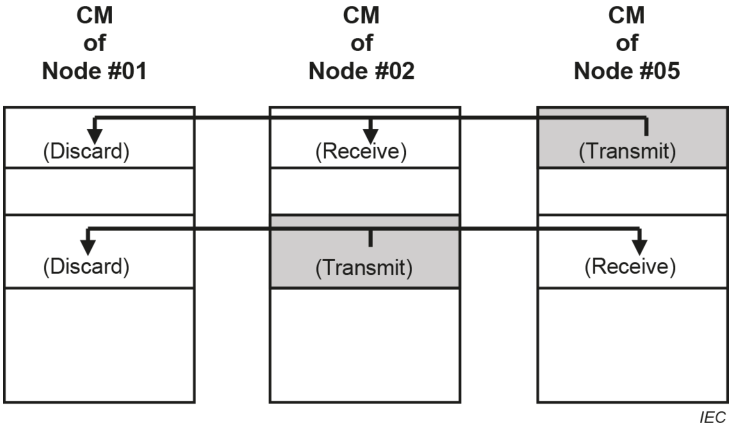


Figure 7 – Node #01 without the CM

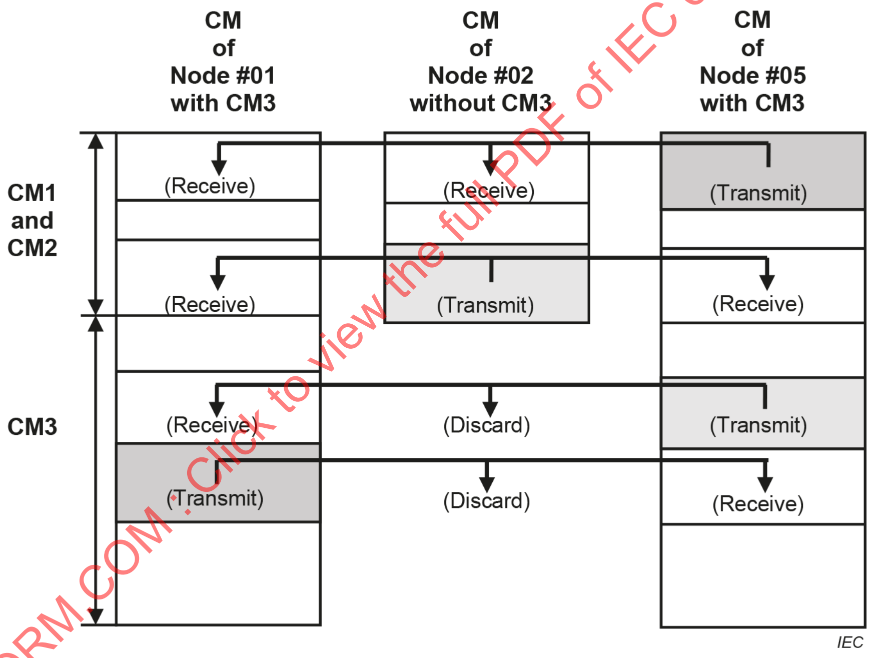


Figure 8 – Data sharing among nodes with and without CM3

6.4.4 CM data type

The data type applied to the CM is primarily based on the basic data type defined in Clause 5. Using these data types, Array or Structure is built and applied.

6.5 ASEs

6.5.1 Overview of Type 26 ASEs

The FAL ASEs and their architectural relationships for Type 26 FAL are shown in Figure 9.

Each ASE provides the services as follows:

a) The cyclic-data ASE provides the cyclic data communication service with Type 26 specific common-memory defined in 6.4, and the cyclic-data transmission is performed at a constant rate for updating the common-memory;

b) Message data ASE provides the message-data communication service between nodes and the message-data transmission is performed on demand by the ALS-user in order to intercommunicate between the APs running on remote nodes;

c) The load measurement ASE provides services to conduct the start and the termination of the communication load measurement on the target node, and provides the measurement result containing the token holding time related and the general purpose communication load related statistical information on that node available for analyzing the total communication load condition over the Type 26 fieldbus network;

d) The General-purpose command server ASE provides the general purpose command server communication service, and provides the facilities for setting and reading the values of various network visible APOs of other nodes, and for requesting the start and the stop of the communication load measurement to other nodes;

e) The Network management ASE provides services to access Type 26 APO attributes; modify the APO instances; configure the nodes; inform the status report on the unexpected events, the errors and the status changes of the nodes; manage network visible APOs accessed through the FAL.

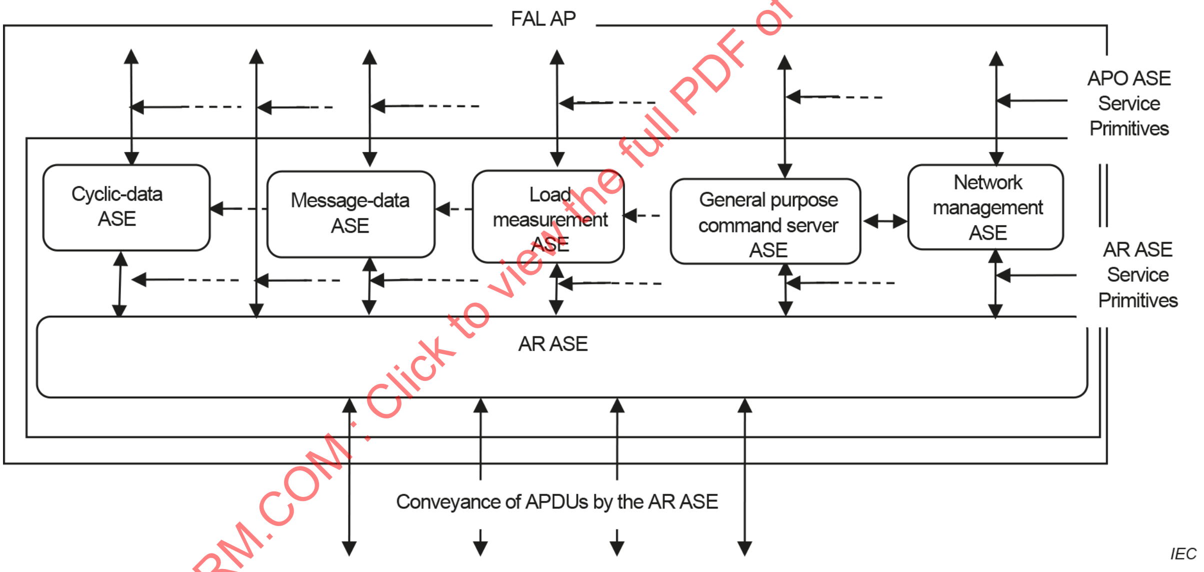


Figure 9 – The structure of ASEs for Type 26 FAL

6.5.2 Type 26 specific conventions for FAL service common parameters

The following Type 26 FAL service parameters are used commonly in the subclauses in which the corresponding services are defined. The definitions for these service parameters are provided herein, and are referred without the definitions throughout the remainder of this document, otherwise extended definitions and/or additional information are provided to these parameters in each case.

Argument
This parameter conveys the service specific parameters of the service request.

DNA
This parameter specifies the destination-node-number.

SNA
This parameter indicates the source-node-number.

TCD
This attribute specifies the transaction identification to be used to convey the service and to identify the context of the application relationship as well as the AREP.

AREP
This attribute indicates information for local identification of the AREP to be used to convey the service and to identify the AR.

Result (+)
This selection type parameter conveys the service specific parameters of the service response, and indicates that the service request is successfully carried out.

M_RLT
This parameter contains the message result by the responder. The value is:

– "Success – Successfully completed".

Result (–)
This selection type parameter conveys the service specific parameters of the service response, and indicates that the service request is failed.

Error info
This parameter conveys the error information on Result (–) of the service response. The value contains:

– "Failure – invalid requested parameter";
– "Failure – service unimplemented";
– "Failure – with other reason".

6.5.3 Cyclic-data ASE

6.5.3.1 Overview

The cyclic-data ASE represents the cyclic data communication service with common-memory in 6.4, and the cyclic-data transmission is performed at a constant rate for updating the contents of the CM.

6.5.3.2 Common memory model class specification

6.5.3.2.1 Overview

The CM ASE provides the write and read data services to/from the CM for the ALS-user in order to intercommunicate between the APs running on remote nodes.

6.5.3.2.2 Common memory class formal model

FAL ASE:			**CM ASE**
CLASS:			**CM**
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1	(o)	Key Attribute:	not used
2	(o)	Key Attribute:	TCD
3	(m)	Attribute:	State
4	(m)	Attribute:	AREP
5	(m)	Attribute:	Network address
6	(m)	Attribute:	Node number
7	(m)	Attribute:	CM (Area 1, Area 2, Area 3)
8	(m)	Attribute:	CM_SZ (Size 1, Size 2, Size 3)
9	(m)	Attribute:	Node status
SERVICES:			
1	(m)	OpsService:	Write
2	(m)	OpsService:	Send-CM
3	(m)	OpsService::	Read
4	(m)	OpsService:	Update-memory
5	(m)	OpsService:	Get-buffer

6.5.3.2.3 Attributes

TCD
This attribute specifies the transaction identification to be used to convey the service and to identify the context of the application relationship as well as the AREP.

State
This attribute indicates the current state of the CM ASE (FSPM) that is defined in detail in IEC 61158-6-26.

AREP
This attribute indicates information for local identification of the AREP to be used to convey the service and to identify the AR.

Network address
This attribute identifies the network address of the node.

Node number
This attribute identifies the node number of the node.

CM (Area 1, Area 2, Area 3)
This attribute identifies the common-memory-area-1, -2 and -3.

CM_SZ (Size 1, Size 2, Size3)
This attribute indicates the data-sizes of the common-memory-area-1, -2 and -3.

Node status
This attribute identifies the node status of this node.

6.5.3.2.4 Services

Write
The Write request primitive is used by the ALS-user to update the content of the Common memory.

Send-CM
The Send-CM request primitive is used by the ALS-user to put the updated local Common memory data on the designated memory area to the send buffer and to send out the data out of the buffer as a publisher.

Read
The Read request primitive is used by the ALS-user to read the content of the Common memory.

Update memory
The Update Memory indication primitive is used to notify the ALS-user that the designated Common Memory area is updated.

Get-buffer
The Get-buffer request primitive is used by the ALS-user to get the data out of the receive buffer that contains the updated Common memory data from a Push-publisher, and to update with this data the corresponding local Common memory of the subscriber.

6.5.3.3 Service specification

6.5.3.3.1 Supported services

The services provided by the CM ASE are as follows:

- Write,
- Send-CM,
- Read,
- Update memory,
- Get-buffer.

6.5.3.3.2 Write

6.5.3.3.2.1 Service overview

The Write service is used by the ALS-user to update the content of the CM.

6.5.3.3.2.2 Service primitives

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

Table 1 – Write service parameters

Parameter name	Req	Cnf
Argument		
C_AD	M	
C_SZ	M	
Memory contents	M	
Result (+)		S
Result (–)		S
Error Info		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

C_AD
This attribute specifies the data-head-address on the common-memory area.

C_AD1
The data-head-address on common-memory-area-1. C_AD1 is specified in combination with C_SZ1.

C_AD2
The data-head-address on common-memory-area-2. C_AD2 is specified in combination with C_SZ2.

C_AD3
The data-head-address on common-memory-area-3. C_AD3 is specified in combination with C_SZ3.

C_SZ
C_SZ specifies the followings.

C_SZ1
The data-size on common-memory-area-1. C_SZ1 is specified in combination with C_AD1.

C_SZ2
The data-size on common-memory-area-2. C_SZ1 is specified in combination with C_AD1.

C_SZ3
The data-size on common-memory-area-3. C_SZ3 is specified in combination with C_AD3.

Memory contents
This parameter specifies the memory contents to write.

6.5.3.3.3 Send-CM

6.5.3.3.3.1 Service overview

The Send-CM service is used by the ALS-user to put the updated local Common memory data on the designated memory area to the send buffer and to send the data out of the buffer as a publisher.

6.5.3.3.3.2 Service primitives

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

Table 2 – Send-CM service parameters

Parameter name	Req	Cnf
Argument		
TCD	M	M (=)
AREP	M	M
C_AD	M	M (=)
C_SZ	M	M (=)
Memory contents	M	
Result (+)		S
Result (–)		S
Error Info		M

C_AD
Refer to 6.5.3.3.2.2.

C_SZ
Refer to 6.5.3.3.2.2.

6.5.3.3.4 Read

6.5.3.3.4.1 Service overview

The Read service is used by the ALS-user to read the content of the CM.

6.5.3.3.4.2 Service primitives

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

Table 3 – Read service parameters

Parameter name	Req	Cnf
Argument		
C_AD	M	
C_SZ	M	
Result (+)		S
Memory contents		M
Result (–)		S
Error Info		M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

C_AD
Refer to 6.5.3.3.2.2.

C_SZ
Refer to 6.5.3.3.2.2.

Memory contents
This parameter is the contents read from the common memory.

6.5.3.3.4.3 Service procedure

The ALS-user issues Read request primitive to the CM ASE, and the ASE returns the memory contents at the designated location on the CM.

6.5.3.3.5 Update memory

6.5.3.3.5.1 Service overview

The Update memory service informs the ALS-user that the contents of the CM at the designated target memory are updated.

6.5.3.3.5.2 Service primitives

The service parameters for this service are shown in Table 4.

Table 4 – Update memory service parameters

Parameter name	Ind
Argument	
C_AD	M
C_SZ	M
Memory status code	M

C_AD
Refer to 6.5.3.3.2.2.

C_SZ
Refer to 6.5.3.3.2.2.

Memory status code
This parameter indicates the status of the CM.

- Contents-updated

6.5.3.3.6 Get-buffer

6.5.3.3.6.1 Service overview

The Get-buffer request primitive is used by the ALS-user to get the data out of the receive buffer that contains the updated Common memory data from a Push-publisher, and to update with this data the corresponding local Common memory of the subscriber.

6.5.3.3.6.2 Service primitives

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

Table 5 – Get- buffer service parameters

Parameter name	Req	Cnf
Argument		
Result (+)		S
C_AD		M
C_SZ		M
Memory status code		M
Memory contents		M
Result (–)		S
Error Info		M

C_AD
Refer to 6.5.3.3.2.2

C_SZ
Refer to 6.5.3.3.2.2

Memory status code
This parameter indicates the status of the CM

Memory contents
This parameter is the contents read from the common memory.

6.5.4 Message data ASE

6.5.4.1 Overview

Message data ASE provides the message-data communication service between nodes in 6.3, and the message-data transmission is performed on demand by the ALS-user in order to intercommunicate between the APs running on remote nodes.

The message-data transmission is of Unconfirmed/Confirmed Client/Server model with Push/Pull types, and supports one to N Unconfirmed Push type as well as one-to-one or Peer type communication service.

6.5.4.2 Message data model class specification

6.5.4.2.1 Message data class formal model

FAL ASE:			**Message-data ASE**
CLASS:			**Message data**
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1	(o)	Key Attribute:	not used
2	(m)	Key Attribute:	TCD
3	(m)	Attribute:	State
4	(m)	Attribute:	AREP
5	(m)	Attribute:	Network address
6	(m)	Attribute:	Node number
7	(m)	Attribute:	Common memory space (Area 1, Area 2, Area 3)
8	(m)	Attribute:	Virtual-address-space (Byte block area, Word block area)
9	(m)	Attribute:	Node status
10	(m)	Attribute:	List of node management information parameters
11	(m)	Attribute:	Network management information
12	(m)	Attribute:	SEQ number list (TX-SEQ, RX-SEQ)
13	(m)	Attribute:	V_SEQ version (TX-V_SEQ, RX- V_SEQ)
14	(m)	Attribute:	Log data memory area
SERVICES:			
1	(o)	OpsService:	Byte block read
2	(o)	OpsService:	Byte block write
3	(o)	OpsService::	Word block read
4	(o)	OpsService:	Word block write
5	(m)	OpsService:	Network parameter read
6	(o)	OpsService:	Network parameter write
7	(o)	OpsService:	Stop command
8	(o)	OpsService:	Operation command
9	(m)	OpsService:	Profile read
10	(o)	OpsService:	Transparent message
11	(m)	OpsService:	Log data read
12	(m)	OpsService:	Log data clear
13	(m)	OpsService:	Message return
14	(o)	OpsService:	Vendor specific message
15	(o)	OpsService:	Set remote node configuration parameter
16	(o)	OpsService:	Read remote participating node management information parameter
17	(o)	OpsService:	Read remote node management information parameter
18	(o)	OpsService:	Read remote node setting information parameter
19	(m)	OpsService:	Extended network parameter read
20	(o)	OpsService:	Extended network parameter write

6.5.4.2.2 Attributes

TCD
This attribute specifies the transaction identification to be used to convey the service and to identify the context of the application relationship as well as the AREP.

State
This attribute indicates the current state of the Message data ASE (FSPM) that is defined in detail in IEC 61158-6-26.

AREP
This attribute indicates information for local identification of the AREP to be used to convey the service and to identify the AR.

Network address
This attribute identifies the network address of the node.

Node number
This attribute identifies the node number of the node.

Common memory space (Area 1, Size 1, Area 2, Size 2, Aria 3, Size 3)
This attribute identifies the common-memory-area-1 and -2.

Virtual-address-space (Byte block area, Word block area)
This attribute identifies the Virtual-address-space of Byte block and Word block areas.

Node status
This attribute identifies the node status of this node.

List of node management information parameters
This attribute identifies the parameters list of the node management information of the nodes.

List of Network management information
This attribute identifies the parameters list of the Network management information of the node.

List of SEQ number (TX-SEQ, RX-SEQ)
This attribute identifies the list of the sequence number for data-transmission and -reception of the nodes.

V_SEQ version (TX-V_SEQ, RX- V_SEQ)
This attribute indicates the version of sequence number for data-transmission and -reception.

Log data memory area
This attribute indicates the memory area information in which the log data is stored.

6.5.4.2.3 Services

Byte block read
The Byte block read request primitive is used to read out the byte-block-data of a target node from other nodes.

Byte block write
The Byte block write request primitive is used to write the byte-block-data to a target node from other nodes.

Word block read
The Word block read request primitive is used to read out the Word-block-data of a target node from other nodes.

Word block write
The Word block write request primitive is used to write the Word-block-data to a target node from other nodes.

Network parameter read
The Network parameter read request primitive is used to retrieve the Network parameters of a target node from other nodes.

Extended network parameter read
The Extended network parameter read request primitive is used to retrieve the extended network parameters of a target node from other nodes.

Network parameter write
The Network parameter write request primitive is used to write the Network parameters to a target node from other nodes.

Extended network parameter write
The Extended network parameter write request primitive is used to write the extended network parameters to a target node from other nodes.

Stop command
The Stop command request primitive is used to set a target node to the STOP state from other nodes.

Operation command
The Operation command request primitive is used to set a target node to the OPERATION state from other nodes.

Profile read
The Profile read request primitive is used to retrieve the Profile data of a target node from other nodes.

Transparent message
The Transparent message request primitive is used to send out a user-defined-service message to other nodes.

Log data read
The Log data read request primitive is used to retrieve the Log data of a target node from other nodes.

Log data clear
The Log data clear request primitive is used to clear the Log data of a target node from other nodes.

Message return
The message return request primitive is used to conduct the message-data transmission test with a procedure that the target node sends back the received message data to the initiating node.

Vendor specific message
The Vendor specific message request primitive is used to request the vendor specific message to other nodes.

Set remote node configuration parameter
The Set remote node configuration parameter request primitive is used to set the node configuration parameters to other nodes.

Read remote participating node management information parameter
The Read remote participating node management information parameter request primitive is used to read the participating node management information parameters of other nodes.

Read remote node management information parameter
The Read remote node management information parameter request primitive is used to retrieve the node management information parameters of other nodes.

Read remote node setting information parameter
The Read remote node setting information parameter request primitive is used to retrieve the node setting information parameters of other nodes.

6.5.4.3 Service specification

6.5.4.3.1 Supported services

The services provided by the Message data ASE are as follows:

- Byte block read service,
- Byte block write service,
- Word block read service,
- Word block write service,
- Network parameter read service,
- Extended network parameter read service,
- Network parameter write service,
- Extended network parameter write service,
- Stop command service,
- Operation command service,
- Profile read service,
- Transparent message service,
- Log data read service,
- Log data clear service,
- Message return service,
- Vendor specific message service,
- Set remote node configuration parameter,
- Read participating remote node management information parameter,
- Read remote node management information parameter,
- Read remote node setting information parameter.

6.5.4.3.2 Byte block read service

6.5.4.3.2.1 Service overview

The Byte block read service is used by the ALS-user to read out the byte-block-data of a target node.

The byte-block-data is on the virtual-address-space which is a virtual memory space composed of BYTE/Unsigned8 data array, and of which element is accessible with the pseudo logical 32-bit address. Figure 10 shows the virtual-address-space.

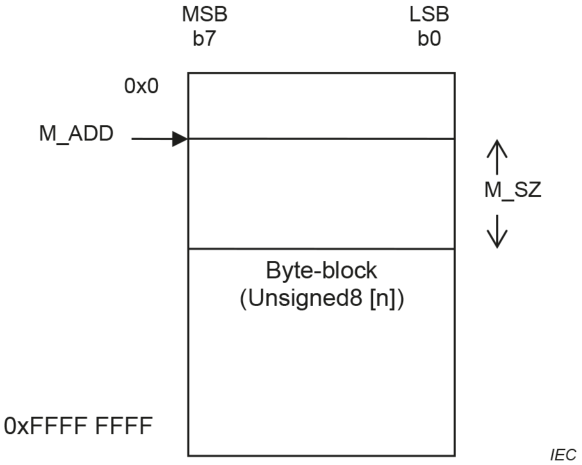


Figure 10 – Virtual-address-space for Byte block

6.5.4.3.2.2 Service primitives

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

Table 6 – Byte block read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
M_ADD	M	M (=)	M (=)	M (=)
M_SZ	M	M (=)	M (=)	M (=)
Result (+)			S	S (=)
Data			M	M (=)
M_RLT			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. See 1.2.				

M_ADD
This parameter specifies the offset address of the target byte block.

M_SZ
This parameter specifies the size of the target byte block.

Data
This parameter contains the value of the target byte block.

6.5.4.3.3 Byte block write service

6.5.4.3.3.1 Service overview

The Byte block write service is used by the ALS-user to write the byte-block-data to the virtual-address-space on a target node. The virtual-address-space is defined in 6.5.4.3.2.

6.5.4.3.3.2 Service primitives

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

Table 7 – Byte block write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
M_ADD	M	M (=)	M (=)	M (=)
M_SZ	M	M (=)	M (=)	M (=)
Data	M	M (=)		
Result (+)			S	S (=)
M_RLT			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. See 1.2.

M_ADD
This parameter specifies the offset address of the target byte block.

M_SZ
This parameter specifies the size of the target byte block.

Data
This parameter contains the value to the target byte block.

6.5.4.3.4 Word block read service

6.5.4.3.4.1 Service overview

The Word block read service is used by the ALS-user to read out the Word-block-data of a target node.

The Word-block-data is on the virtual-address-space which is a virtual memory space composed of WORD/Unsigned16 data array, and of which element is accessible with the pseudo logical 32-bit address. Figure 11 shows the virtual-address-space.

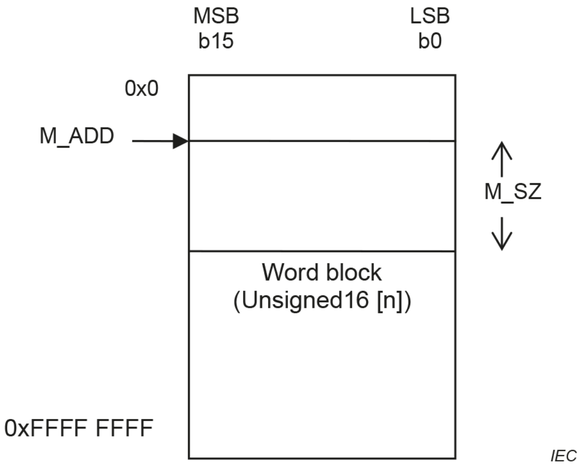


Figure 11 – Virtual-address-space for Word block

6.5.4.3.4.2 Service primitives

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

Table 8 – Word block read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
M_ADD	M	M (=)	M (=)	M (=)
M_SZ	M	M (=)	M (=)	M (=)
Result (+)			S	S (=)
Data			M	M (=)
M_RLT			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. See 1.2.

M_ADD
This parameter specifies the offset address of the target word block.

M_SZ
This parameter specifies the size of the target word block.

Data
This parameter contains the value of the target word block.

6.5.4.3.5 Word block write service

6.5.4.3.5.1 Service overview

The Word block write service is used by the ALS-user to write the Word-block-data to the virtual-address-space of a target node. The virtual-address-space is defined in 6.5.4.3.4.

6.5.4.3.5.2 Service primitives

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

Table 9 – Word block write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
M_ADD	M	M (=)	M (=)	M (=)
M_SZ	M	M (=)	M (=)	M (=)
Data	M	M (=)		
Result (+)			S	S (=)
M_RLT			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. See 1.2.

M_ADD
This parameter specifies the offset address of the target word block.

M_SZ
This parameter specifies the size of the target word block.

Data
This parameter contains the values to the target word block.

6.5.4.3.6 Network parameter read service

6.5.4.3.6.1 Service overview

The Network parameter read service is used by the ALS-user to retrieve the Network parameters of a target node.

6.5.4.3.6.2 Service primitives

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

Table 10 – Network parameter read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			M	M (=)
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. See 1.2.				

Data
This parameter contains the values of the target network parameters read from the target node.

Node name of the node (NDN)
This parameter indicates the node name by the end user.

Vendor code of the node (VDN)
This parameter indicates the vendor code of the node by the vendor.

Manufacturer model name of the node (MSN)
This parameter indicates the manufacturer model name of the node.

Area 1 data head address (C_AD1)
This parameter indicates the data-head-address on common-memory-area-1.

Area 1 data size (C_SZ1)
This parameter indicates the data-size on common-memory-area-1.

Area 2 data head address (C_AD2)
This parameter indicates the data-head-address on common-memory-area-2.

Area 2 data size (C_SZ2)
This parameter indicates the data-size on common-memory-area-2.

Token watchdog time (TW)
This parameter indicates the observation time period for the token circulation.

Allowable minimum frame interval time (MFT)
This parameter indicates the allowable-minimum-frame-interval-time for the message-data transmission.

FA link status (LKS)
This parameter indicates the link-status.

Protocol type (P_TYPE)
This parameter indicates the protocol-type.

Upper layer status (ULS)
This parameter indicates the upper-layer status.

Allowable refresh cycle time setting value (RCT)
This parameter indicates the value of the allowable-refresh-cycle-time.

Refresh cycle measurement time–current value (RMT_current)
This parameter indicates the current measurement value of the refresh-cycle-time.

Refresh cycle measurement time–maximum value (RMT_maximum)
This parameter indicates the maximum measurement value of the refresh-cycle-time.

Refresh cycle measurement time–minimum value (RMT_minimum)
This parameter indicates the minimum measurement value of the refresh-cycle-time

6.5.4.3.7 Extended network parameter read service

6.5.4.3.7.1 Service overview

The extended network parameter read service is used by the ALS-user to retrieve the extended network parameters of a target node.

6.5.4.3.7.2 Service primitives

The service parameters for this service are shown in Table 11.

Table 11 – Extended network parameter read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			M	M (=)
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. See 1.2.				

Data
This parameter contains the values of the target extended network parameters read from the target node.

Area 3 data head address (C_AD3)
This parameter indicates the data-head-address on CM3.

Area 3 data size (C_SZ3)
This parameter indicates the data-size on CM3.

6.5.4.3.8 Network parameter write service

6.5.4.3.8.1 Service overview

The Network parameter write service is used by the ALS-user to write the Network parameters to a target node.

6.5.4.3.8.2 Service primitives

The service parameters for this service are shown in Table 12.

Table 12 – Network parameter write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Data	M	M (=)		
Result (+)			S	S (=)
M_RLT			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. See 1.2.

Data

This parameter contains the value of the network parameters to the target node.

Setting parameter flags

This parameter designates the options.

- 0x01: Set only both of the data-head-address and the data-size for CM area-1 and -2;
- 0x02: Set only the node name (NDN);
- 0x03: Set all parameters.

Area 1 data head address (C_AD1)

The data-head-address on common-memory-area-1. C_AD1 is specified in combination with C_SZ1.

Area 1 data size (C_SZ1)

The data-size on common-memory-area-1. C_SZ1 is specified in combination with C_AD1.

Area 2 data head address (C_AD2)

The data-head-address on common-memory-area-2. C_AD2 is specified in combination with C_SZ2.

Area 2 data size (C_SZ2)

The data-size on common-memory-area-2. C_SZ1 is specified in combination with C_AD1.

Node name of the node

This parameter indicates the node name by the end user.

6.5.4.3.9 Extended network parameter write service

6.5.4.3.9.1 Service overview

The Extended network parameter write service is used by the ALS-user to write the extended network parameters to a target node.

6.5.4.3.9.2 Service primitives

The service parameters for this service are shown in Table 13.

Table 13 – Extended network parameter write service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Data	M	M (=)		
Result (+)			S	S (=)
M_RLT			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. See 1.2.				

Data
This parameter contains the value of the extended network parameters to the target node.

Setting parameter flag
This parameter designates the option.

– 0x01: Set only both of the data-head-address and the data-size on CM3.

Area 3 data head address (C_AD3)
The data-head-address on CM3. C_AD3 is specified in combination with C_SZ3.

Area 3 data size (C_SZ3)
The data-size on CM3. C_SZ3 is specified in combination with CAD3.

6.5.4.3.10 Stop command service

6.5.4.3.10.1 Service overview

The Stop command service is used by the ALS-user to set a target node to the STOP state.

6.5.4.3.10.2 Service primitives

The service parameters for this service are shown in Table 14.

Table 14 – Stop command service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			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. See 1.2.

6.5.4.3.11 Operation command service

6.5.4.3.11.1 Service overview

The Operation command service is used by the ALS-user to set a target node to the OPERATION state.

6.5.4.3.11.2 Service primitives

The service parameters for this service are shown in Table 15.

Table 15 – Operation command service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			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. See 1.2.

6.5.4.3.12 Profile read service

6.5.4.3.12.1 Service overview

The Profile read service is used by the ALS-user to retrieve the Profile data of a target node.

The profile data to be retrieved is the device profile data for the devices of other nodes to interconnect and to be interoperable with other devices over the Type 26 fieldbus network.

NOTE The contents of the device profile data depends on each device and the device's vendor.

6.5.4.3.12.2 Service primitives

The service parameters for this service are shown in Table 16.

Table 16 – profile read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			M	M (=)
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. See 1.2.

Data
This parameter contains the values of the device profile data for the device of the target node.

Network parameters
This parameter contains the values of the target network parameters read from the target node.

Node name of the node (NDN)
This parameter indicates the node name by the end user.

Vendor code of the node (VDN)
This parameter indicates the vendor code of the node by the vendor.

Manufacturer model name of the node (MSN)
This parameter indicates the manufacturer model name of the node.

Area 1 data head address (C_AD1)
This parameter indicates the data-head-address on common-memory-area-1.

Area 1 data size (C_SZ1)
This parameter indicates the data-size on common-memory-area-1.

Area 2 data head address (C_AD2)
This parameter indicates the data-head-address on common-memory-area-2.

Area 2 data size (C_SZ2)
This parameter indicates the data-size on common-memory-area-2.

Token watchdog time (TW)
This parameter indicates the observation time period for the token circulation.

Allowable minimum frame interval time (MFT)
This parameter indicates the allowable-minimum-frame-interval-time for the message-data transmission.

FA link status (LKS)
This parameter indicates the link-status.

Protocol type (P_TYPE)
This parameter indicates the protocol-type.

Upper layer status (ULS)
This parameter indicates the upper-layer status.

Allowable refresh cycle time setting value (RCT)
This parameter indicates the value of the allowable-refresh-cycle-time.

Refresh cycle measurement time–current value (RMT_current)
This parameter indicates the current measurement value of the refresh-cycle-time.

Refresh cycle measurement time–maximum value (RMT_maximum)
This parameter indicates the maximum measurement value of the refresh-cycle-time.

Refresh cycle measurement time–minimum value (RMT_minimum)
This parameter indicates the minimum measurement value of the refresh-cycle-time

System parameters
This parameter represents the management information to identify the device as follows:

Common parameters
This parameter identifies the followings:

- Version of the device profile common specification;
- System parameter ID;
- System parameter revision number;
- System parameter revision date;
- Device type;
- Vendor name;
- Product name.

Device specific parameters
This parameter identifies the device specific item.

- Device-specific parameter ID

Communication object information
This parameter represents the object information with which other devices can access the resource objects within the device.

Information on resource object
This parameter represents the connection identification between the resource object and the virtual address to be used for the Byte block read/write and the Word block read/write services provided with the message-data ASE.

Information on status and mode
This parameter identifies which status exists, how the status is visible, whether and how the mode is controllable in relation to the Stop/Start services provided with the message-data ASE.

Information on downloading and uploading
This parameter identifies which program or data is uploadable and downloadable, and how they are uploaded or downloaded.

Implementation of message services
This parameter identifies which service of the message-data ASE is applicable.

Information on the transparent message service
This parameter identifies whether and how the transparent message service is provided.

Information on security
This parameter identifies whether and how the security measure is provided.

Implementation of the log data read service
This parameter identifies which measurement item is provided and retrievable with the communication load measurement service.

6.5.4.3.13 Transparent message service

6.5.4.3.13.1 Service overview

The Transparent message service is used by the ALS-user to send a user-defined-service message to a target node. The ALS-user of other nodes shall send out the response using the Transparent message service according to their pre-defined procedure.

6.5.4.3.13.2 Service primitives

The service parameters for this service are shown in Table 17.

Table 17 – Transparent message service parameters

Parameter name	Req	Ind	Cnf
Argument			
DNA	M		
SNA	M	M (=)	
TCD	M	M (=)	
AREP	M	M	
Transmission mode	M	M (=)	
Data	U	U (=)	
Result (+)			S
M_RLT			M
Result (–)			S
Error info			M
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.			

Transmission mode
This parameter specifies the transmission mode; i.e. Point-to-Point (PTP) or Broadcast (BCT).

Data
This parameter contains the user data.

6.5.4.3.14 Log data read service

6.5.4.3.14.1 Service overview

The Log data read service is used by the ALS-user to retrieve the Log data of a target node.

6.5.4.3.14.2 Service primitives

The service parameters for this service are shown in Table 18.

Table 18 – Log data read service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M (=)	M	
Result (+)			S	S
M_RLT			M	M
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. See 1.2.				

Data
This parameter contains the value of the Log data transferred from the target node.

Table 19 shows the log data items.

Table 19 – Log data items

Data item	Value	Data type	Usage
Total count of transmission over the TSAP	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of transmission error over the TSAP	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of transmission error over the Ethernet	Roll-over binary counter value of 32-bit length	UDINT	o
Additional information 1	Related additional information	UDINT[3]	o
Total count of reception over the TSAP	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of reception error over the TSAP	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of reception error over the Ethernet	Roll-over binary counter value of 32-bit length	UDINT	o

Data item	Value	Data type	Usage
Additional information 2	Related additional information	UDINT[3]	o
Total count of the token sent out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the cyclic-data frame sent out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the one-to-one message-data frame sent out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the one-to-n message-data frame sent out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the extended-cyclic-data frame sent out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Additional information 3	Related additional information	UDINT	o
Total count of the token frame received since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the cyclic-data frame received since the measurement start	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the one-to-one message-data frame received since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the one-to-n message-data frame received since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the extended-cyclic-data frame received since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Additional information 4	Related additional information	UDINT	o
Total count of errors in receiving the cyclic-data frame since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of the cyclic-data frame received with CM address size error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the cyclic-data frame received with fragment number error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the cyclic-data frame received with total fragment count error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the cyclic-data frame received with block size error since the measurement starts	Roll-over binary counter value of 32-bit length.	UDINT	o
Elapsed time since the measurement starts when the error is detected in receiving cyclic-data frame	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Total count of errors in receiving the extended-cyclic-data frame since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the extended-cyclic-data frame received with CM3 address size error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o

Data item	Value	Data type	Usage
Total count of the extended-cyclic-data frame received with fragment number error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the extended- cyclic-data frame received with total fragment count error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the extended-cyclic-data frame received with block size error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Elapsed time since the measurement starts when the error is detected in receiving extended-cyclic-data frame	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Total count of the message-data frame retransmitted since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of the message-data frame retransmitted over the retry counts since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m
Elapsed time since the measurement starts when the maximum refresh-cycle time occurred last	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Additional information 6	Related additional information	UDINT[3]	o
Total count of errors in reception of the message-data frame since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of the message-data frame received with sequence number error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the message-data frame retransmitted with same sequence number since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Elapsed time since the measurement starts when receiving message-data frame is in error	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Additional information 7	Related additional information	UDINT[2]	o
Total count of the ACK error since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of the ACK error with wrong version number of sequence number since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the ACK error with wrong sequence number since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the ACK error with wrong node number since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the ACK error with wrong TCD since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Additional information 8	Related additional information	UDINT[7]	o
Total count of the duplicated token frame error since the measurement start	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of the token discarded since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m

Data item	Value	Data type	Usage
Total count of the token reissued since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	m
Elapsed time since the measurement starts when the token is discarded	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Elapsed time since the measurement starts when the token reissued last	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Total count of the token holding time out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Total count of the token-watchdog-time out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT	o
Elapsed time since the measurement starts when the token watchdog time out occurred last	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Maximum time of the token holding time since the measurement starts	μs Roll-over binary counter value of 32-bit length. The unit is 1 μs.	UDINT	o
Minimum time of the token holding time since the measurement starts	μs Roll-over binary counter value of 32-bit length. The unit is 1 μs.	UDINT	o
Elapsed time since the measurement starts when the maximum token holding time occurred last	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Elapsed time in operation state	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Elapsed time since the measurement starts when the token holding time out occurred last	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Total count of being in waiting frame reception state	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of entering in participating state	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of entering in dropping out state of the node	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of entering in dropping out state with the token skipped	Roll-over binary counter value of 32-bit length	UDINT	m
Total count of entering in dropping out state of other nodes	Roll-over binary counter value of 32-bit length	UDINT	m
Elapsed time since the measurement starts of the token-holding-time measurement	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Total count of the token frames received during the measurement of the token-holding time	Roll-over binary counter value of 32-bit length	UDINT	o
Reserved	0x0	UDINT[3]	–
Measurement time elapsed since the sender log measurement starts	sec. Roll-over binary counter value of 32-bit length. The unit is 1 s.	UDINT	o
Participation node related information	Related data on participation nodes	UDINT[n]	o
IP 1	Sender IP address 1	BitString32	o
Number of the frame reception from IP 1	Roll-over binary counter value of 32-bit length	UDINT	o
IP 2	Sender IP address 2	BitString32	o

Data item	Value	Data type	Usage
Number of the frame reception from IP 2	Roll-over binary counter value of 32-bit length	UDINT	o
IP 3	Sender IP address 3	BitString32	o
Number of the frame reception from IP 3	Roll-over binary counter value of 32-bit length	UDINT	o
IP 4	Sender IP address 4	BitString32	o
Number of the frame reception from IP 4	Roll-over binary counter value of 32-bit length	UDINT	o
IP 5	Sender IP address5	BitString32	o
Number of the frame reception from IP 5	Roll-over binary counter value of 32-bit length	UDINT	o
IP 6	Sender IP address 6	BitString32	o
Number of the frame reception from IP 6	Roll-over binary counter value of 32-bit length	UDINT	o
IP 7	Sender IP address 7	BitString32	o
Number of the frame reception from IP 7	Roll-over binary counter value of 32-bit length	UDINT	o
IP 8	Sender IP address 8	BitString32	o
Number of the frame reception from IP 8	Roll-over binary counter value of 32-bit length	UDINT	o
IP 9	Sender IP address 9	BitString32	o
Number of the frame reception from IP 9	Roll-over binary counter value of 32-bit length	UDINT	o
IP 10	Sender IP address 10	BitString32	o
Number of the frame reception from IP 10	Roll-over binary counter value of 32-bit length	UDINT	o
Vendor specific items	Items dependent on implementation	UDINT[n]	-
NOTE In the usage column, "m" denotes "mandatory" and "o" denotes "optional".			

6.5.4.3.15 Log data clear service

6.5.4.3.15.1 Service overview

The Log data clear service is used by the ALS-user to clear the Log data of a target node.

6.5.4.3.15.2 Service primitives

The service parameters for this service are shown in Table 20.

Table 20 – Log data clear service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		C	
SNA	M	M (=)	C	M
TCD	M	M (=)	C	
AREP	M	M	C	
Transmission mode	M	M (=)	C	
Result (+)			C	S
M_RLT			C	M
Result (–)			C	S
Error info			C	M

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

Transmission mode
This parameter specifies the transmission mode:

- Point-to-Point (PTP);
- Broadcast (BCT).

6.5.4.3.16 Message return service

6.5.4.3.16.1 Service overview

The Message return service is used by the ALS-user to conduct the message-data transmission test to a target node with a procedure that the target node sends back the received message data to the initiating node.

6.5.4.3.16.2 Service primitives

The service parameters for this service are shown in Table 21.

Table 21 – Message return service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Data	M	M (=)	M (=)	
Result (+)			S	S
M_RLT			M	M
Received 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. See 1.2.

Data
This parameter contains the value of the test data to send out.

Received data
This parameter contains the value of the received data to send back.

6.5.4.3.17 Vendor specific message service

6.5.4.3.17.1 Service overview

The Vendor specific message service is used by the ALS-user to request the vendor specific message to a target node.

6.5.4.3.17.2 Service primitives

The service parameters for this service are shown in Table 22.

Table 22 – Vendor specific message service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		C	
SNA		M	C	M
TCD	M	M (=)	C	
AREP	M	M	C	
Transmission mode	M	M (=)		
VDN	M	M (=)	C	M (=)
SCODE	M	M (=)	C	M (=)
Data	M	M (=)		
Result (+)			C	S (=)
M_RLT			C	M (=)
Data			C	M (=)
Result (–)			C	S (=)
Error info			C	M

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

Transmission mode
This parameter specifies the transmission mode:

- Point-to-Point (PTP);
- Broadcast (BCT).

VDN
This parameter indicates the vendor code of the node.

SCODE
This parameter indicates the vendor specific subcode.

Data
This parameter contains the message to be exchanged.

6.5.4.3.18 Set remote node configuration parameter

6.5.4.3.18.1 Service overview

The Set remote node configuration parameter service is used by the ALS-user to set the node configuration parameters to other nodes.

6.5.4.3.18.2 Service primitives

The service parameters for this service are shown in Table 23.

Table 23 – Set remote node configuration parameter service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Data	M	M (=)		
Result (+)			S	S (=)
M_RLT			M	M (=)
Data values			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. See 1.2.

Data
This parameter contains the values to be set to the node configuration parameters of a target node.

Table 24 shows the data elements and the node configuration parameters to be set to the target node.

Table 24 – Data elements and Node configuration parameters

Parameter name	Description	Value	Data type

Update flag [a]	Flag to control the update of each parameter	– C_AD 1, C_AD2, C_SZ1 and C_SZ2: Update or Not;
		– NDN: Update or NOT;
		– TW: Update or NOT;
		– MFT: Update or NOT;
			BitString16

NDN	Node name by the end user	ASCII character string	char[10]
C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT
TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	USINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 µs.	USINT

[a] This parameter is conveyed to the target node and is not one of the parameters of the node configuration parameters.

Data values
This parameter contains the parameter values completed to set up to the node configuration parameters of the target node.

6.5.4.3.19 Read remote participating node management information parameter

6.5.4.3.19.1 Service overview

The Read remote participating node management information parameter service is used by the ALS-user to retrieve the participating node management information parameters of other nodes.

6.5.4.3.19.2 Service primitives

The service parameters for this service are shown in Table 25.

Table 25 – Service parameters of Read remote participating node management information parameter service

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Node number	M	M (=)		
Result (+)			S	S (=)
M_RLT			M	M (=)
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. See 1.2.

Node number
This parameter specifies the target node number.

Data
This parameter contains the values of the participating node management information parameters transferred from the target node.

Table 26 shows the participating node management information parameters transferred from the target node.

Table 26 – Participating node management information parameters

Parameter name	Description	Value	Data type
Node_number	Node number	1 to 254	UINT

ULS	Upper layer status consists of the operating state and the error information.	– Operating state: "RUN" or "STOP"
		– Error information:
		– "NORMAL" or NOT;
		– "WARNING" or NOT;
		– "ALARM" or NOT;
		– Error_ID
			BitString16

C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT
RCT	Allowable-refresh-cycle-time	0 to 65 535. The unit is 1 ms.	UINT
TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	UINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 μs.	UINT

LKS	FA link status consists of:
	– "Overlapped_CM_area" state flag;
	– "CM_Set_up_Comp" state flag;
	– "CM_Data_Valid" state flag;
	– "Upper_Layer_inactivity" state flag;
	– "Communication_invalid" state flag;
	– "In_Ring" state flag.
		– "Overlapped_CM_area" or NOT;
		– "CM_Set_up_Comp" or NOT;
		– "CM_Data_Valid" or NOT;
		– "Upper_Layer_inactivity" or NOT;
		– "Communication_invalid" or NOT;
		– "In_Ring" or NOT.
			BitString16

6.5.4.3.20 Read remote node management information parameter

6.5.4.3.20.1 Service overview

The Read remote node management information parameter service is used by the ALS-user to retrieve the node management information parameters of other nodes.

6.5.4.3.20.2 Service primitives

The service parameters for this service are shown in Table 27.

Table 27 – Read remote node management information parameter service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			M	M (=)
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. See 1.2.				

Data
This parameter contains the values of the node management information parameters transferred from the target node.

Table 28 shows the node management information parameters transferred from the target node.

Table 28 – Node management information parameters

Parameter name	Description	Value	Data type
Node_number	Node number	1 to 254	UINT
C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT

ULS	Upper layer status consists of the operating state and the error information.	– Operating state: "RUN" or "STOP"
		– Error information:
		– "NORMAL" or NOT;
		– "WARNING" or NOT;
		– "ALARM" or NOT;
		– Error_ID
			BitString16

TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	UINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 µs.	UINT
VDN	Vendor code of the node	ASCII character string	char[10]
MSN	Manufacturer model name of the node	ASCII character string	char[10]
NDN	Node name by the end user	ASCII character string	char[10]
P_TYPE	Protocol type	0x80: Fixed value	UINT

LKS	FA link status consists of:
	– "Overlapped_CM_area" state flag;
	– "CM_Set_up_Comp" state flag;
	– "CM_Data_Valid" state flag;
	– "Upper_Layer_inactivity" state flag;
	– "Communication_invalid" state flag;
	– "In_Ring" state flag.
		– "Overlapped_CM_area" or NOT;
		– "CM_Set_up_Comp" or NOT;
		– "CM_Data_Valid" or NOT;
		– "Upper_Layer_inactivity" or NOT;
		– "Communication_invalid" or NOT
		– "In_Ring" or NOT.
			BitString16

Node_status	Status of this node consists of:
	– "Duplicate_Node_address" state flag;
	– "Overlapped_CM_area" state flag;
	– "TW_timer_expired" state flag;
	– "Wait_for_recv" state flag;
	– "Invalid_Initialize_Para" state flag.
		– "Duplicate_Node" or "NOT";
		– "Overlapped_CM_area" or "NOT";
		– "TW_timer_expired" or "NOT";
		– "Wait_for_recv" or "NOT";
		– "Invalid_Initialize_Para" or "NOT".
			BitString16

Node_class	Node class identification	0x3: Fixed value	UINT

6.5.4.3.21 Read remote node setting information parameter

6.5.4.3.21.1 Service overview

The Read remote node setting information parameter service is used by the ALS-user to retrieve the node setting information parameters of other nodes.

6.5.4.3.21.2 Service primitives

The service parameters for this service are shown in Table 29.

Table 29 – Read remote node setting information parameter service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
M_RLT			M	M (=)
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. See 1.2.				

Data
This parameter contains the values of the node setting information parameters transferred from the target node.

Table 30 shows the node setting information parameters transferred from the target node.

Table 30 – Node setting information parameters

Parameter name	Description	Value	Data type
C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT
TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	UINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 µs.	UINT
NDN	Node name by the end user	ASCII character string	char[10]

6.5.5 Load measurement ASE

6.5.5.1 Overview

The load measurement ASE provides services to conduct the start and the termination of the communication load measurement on the target node, and provides the measurement result containing the token holding time related and the general purpose communication load related statistical information on that node available for analyzing the total communication load condition over the Type 26 fieldbus network.

6.5.5.2 Load measurement class formal model

FAL ASE:			**Load measurement ASE**
CLASS:			**Load measurement**
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1	(o)	Key Attribute:	not used
2	(o)	Key Attribute:	TCD
3	(m)	Attribute:	State
4	(m)	Attribute:	AREP
4.1	(m)	Attribute:	Type (TCP, UDP)
5	(m)	Attribute:	Network address
6	(m)	Attribute:	Node number
7	(m)	Attribute:	SEQ
8	(m)	Attribute:	Virtual-address-space (Byte block area, Word block area)
9	(m)	Attribute:	Log data memory area
SERVICES:			
1	(m)	OpsService:	Start TK holding time measurement
2	(m)	OpsService:	Terminate TK holding time measurement
3	(m)	OpsService::	Start GP_Comm sender log
4	(m)	OpsService:	Terminate GP_Comm sender log

6.5.5.2.1 Attributes

TCD
This attribute specifies the transaction identification to be used to convey the service and to identify the context of the application relationship as well as the AREP.

State
This attribute indicates the current state of the Message data ASE (FSPM) that is defined in detail in IEC 61158-6-26.

AREP
This attribute indicates information for local identification of the AREP to be used to convey the service and to identify the AR.

Type (TCP, UDP)
This attribute identifies the protocol type of TCP or UDP.

Network address
This attribute identifies the network address of the node.

Node number
This attribute identifies the node number of the node.

SEQ
This attribute represents the sequence number and the value is used to determine whether a receiving message has been previously received and is redundant.

Virtual-address-space (Byte block area, Word block area)
This attribute identifies the Virtual-address-space of Byte block and Word block areas.

Log data memory area
This attribute indicates the memory area information in which the log data is stored.

6.5.5.2.2 Services

Start TK-holding-time measurement
The Start TK-holding-time measurement request primitive is used to request the start of the token-holding-time measurement to the target node.

Terminate TK holding time measurement
The Terminate TK-holding-time measurement request primitive is used to request the termination of the token-holding-time measurement to the target node.

Start GP_Comm sender log
The Start GP_Comm sender log request primitive is used to request the start of the General purpose-communication-sender log measurement to the target node.

Terminate GP_Comm sender log
The Terminate GP_Comm sender log request primitive is used to request the termination of the General purpose-communication-sender log measurement to the target node.

6.5.5.3 Service specification

6.5.5.3.1 Supported services

The service provided by the Load measurement ASE is as follows:

- Start TK-holding-time measurement,
- Terminate TK holding time measurement,
- Start GP_Comm sender log,
- Terminate GP_Comm sender log.

6.5.5.3.2 Start TK-holding-time measurement

6.5.5.3.2.1 Service overview

The Start TK-holding-time measurement service is used by the ALS-user to request the start of the token-holding-time measurement to a target node.

6.5.5.3.2.2 Service primitives

The service parameters for this service are shown in Table 31.

Table 31 – Start TK-holding-time measurement service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	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. See 1.2.				

6.5.5.3.3 Terminate TK holding time measurement

6.5.5.3.3.1 Service overview

The Terminate TK-holding-time measurement service is used by the ALS-user to request the termination of the token-holding-time measurement to a target node.

6.5.5.3.3.2 Service primitives

The service parameters for this service are shown in Table 32.

Table 32 – Terminate TK-holding-time measurement service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
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. See 1.2.				

Data
This parameter contains the measurement result transferred from the target node.

Table 33 shows the measurement result transferred from the target node.

Table 33 – Token-holding-time measurement result

Measurement item	Value	Data type
TK_Discarded_count: Total count of the token discarded since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
TK_Discarded_time: Elapsed time since the measurement starts when the token is discarded	sec. The unit is 1 s.	UDINT
TK_Reissued_count: Total count of the token reissued since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
TK_Reissued_time: Elapsed time since the measurement starts when the token reissued last	sec. The unit is 1 s.	UDINT
TK-hld_timeout_count: Total count of the token holding time out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
TK-hld_timeout_time: Elapsed time since the measurement starts when the token holding time out occurred last	sec. The unit is 1 s.	UDINT
TW_timeout_count: Total count of the token-watchdog-time out since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
TW_timeout_time: Elapsed time since the measurement starts when the token watchdog time out occurred last	sec. The unit is 1 s.	UDINT
Max_TK_hld_time: Maximum time of the token holding time since the measurement starts	μs The unit is 1 μs.	UDINT
Min_TK_hld_time: Minimum time of the token holding time since the measurement starts	μs The unit is 1 μs.	UDINT
Max_time: Elapsed time since the measurement starts when the maximum token holding time occurred last	sec. The unit is 1 s.	UDINT
Mtime: Elapsed time since the measurement starts of the token-holding-time measurement	sec. The unit is 1 s.	UDINT
TK_receive_count: Total count of the token frames received during the measurement of the token-holding time	Roll-over binary counter value of 32-bit length	UDINT
Max_RFC_time: Elapsed time since the measurement starts when the maximum refresh-cycle time occurred last	sec. The unit is 1 s.	UDINT
Cyclic_frame_count: Total count of receiving the cyclic-data frame since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
Cyclic_frame_Error_count: Total count of errors in receiving the cyclic-data frame since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
Cyclic_frame_Error_time: Elapsed time since the measurement starts when the error is detected in receiving cyclic-data frame	sec. The unit is 1 s.	UDINT
Message_frame_count: Total count of errors in reception of the message-data frame since the measurement starts	Roll-over binary counter value of 32-bit length	UDINT
Message_frame_Error_time: Elapsed time since the measurement starts when receiving message-data frame is in error	sec. The unit is 1 s.	UDINT

6.5.5.3.4 Start GP_Comm sender log

6.5.5.3.4.1 Service overview

The Start GP_Comm sender log service is used by the ALS-user to request the start of the GP_Comm sender log measurement to a target node.

6.5.5.3.4.2 Service primitives

The service parameters for this service are shown in Table 34.

Table 34 – Start GP_Comm sender log service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	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. See 1.2.				

6.5.5.3.5 Terminate GP_Comm sender log

6.5.5.3.5.1 Service overview

The Terminate GP_Comm sender log service is used by the ALS-user to request the termination of the GP_Comm sender log measurement to a target node.

6.5.5.3.5.2 Service primitives

The service parameters for this service are shown in Table 35.

Table 35 – Terminate GP_Comm sender log service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	M	M	
Result (+)			S	S (=)
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. See 1.2.				

Data
This parameter contains the measurement result transferred from the target node.

Table 36 shows the measurement result transferred from the target node. The measurement result contains the numbers of reception of the general purpose communication data frames from the senders with the source IP addresses. The GP_Comm sender log service provides the log for the first 10 events of the reception from each sender.

Table 36 – GP_Comm sender log measurement result

Measurement item	Value	Data type
Mtime: Measurement time elapsed since the measurement starts	sec The unit is 1 s.	UDINT
IP 1	Sender IP address 1	BitString32
Number of the frame reception from IP 1	Roll-over binary counter value of 32-bit length	UDINT
IP 2	Sender IP address 2	BitString32
Number of the frame reception from IP 2	Roll-over binary counter value of 32-bit length	UDINT
IP 3	Sender IP address 3	BitString32
Number of the frame reception from IP 3	Roll-over binary counter value of 32-bit length	UDINT
IP 4	Sender IP address 4	BitString32
Number of the frame reception from IP 4	Roll-over binary counter value of 32-bit length	UDINT
IP 5	Sender IP address5	BitString32
Number of the frame reception from IP 5	Roll-over binary counter value of 32-bit length	UDINT
IP 6	Sender IP address 6	BitString32
Number of the frame reception from IP 6	Roll-over binary counter value of 32-bit length	UDINT
IP 7	Sender IP address 7	BitString32
Number of the frame reception from IP 7	Roll-over binary counter value of 32-bit length	UDINT
IP 8	Sender IP address 8	BitString32
Number of the frame reception from IP 8	Roll-over binary counter value of 32-bit length	UDINT
IP 9	Sender IP address 9	BitString32
Number of the frame reception from IP 9	Roll-over binary counter value of 32-bit length	UDINT
IP 10	Sender IP address 10	BitString32
Number of the frame reception from IP 10	Roll-over binary counter value of 32-bit length	UDINT

6.5.6 Network management ASE

6.5.6.1 Overview

The Network management ASE provides services to access Type 26 APO attributes; modify the APO instances; configure the nodes; inform the status report on the unexpected events, the errors and the status changes of the nodes; manage network visible APOs accessed through the FAL.

6.5.6.2 Network management model class specification

6.5.6.2.1 Network management class formal model

FAL ASE:	**Network management ASE**
CLASS:	**NM**
CLASS ID:	not used
PARENT CLASS:	TOP

ATTRIBUTES:

1	(o)	Key Attribute:	not used
2	(o)	Key Attribute:	TCD
3	(m)	Attribute:	State
4	(m)	Attribute:	AREP
5	(m)	Attribute:	Network address
6	(m)	Attribute:	Node number
7	(m)	Attribute:	Common memory space (Area 1, Area 2)
8	(m)	Attribute:	Virtual-address-space (Byte block area, Word block area)
9	(m)	Attribute:	Node status
10	(m)	Attribute:	List of node management information parameters
11	(m)	Attribute:	Network management information
12	(m)	Attribute:	SEQ number list (TX-SEQ, RX-SEQ)
13	(m)	Attribute:	V_SEQ version (TX-V_SEQ, RX- V_SEQ)
14	(m)	Attribute:	Log data memory area

SERVICES:

1	(m)	OpsService:	Set configuration parameter
2	(m)	OpsService:	Read node management information parameter
3	(m)	OpsService::	Read participating node management information parameter
4	(m)	OpsService:	Read network management information parameter
5	(m)	OpsService:	Read message sequence number management information
6	(m)	OpsService:	Read node status
7	(o)	OpsService::	Reset node
8	(o)	OpsService:	Set network address
9	(m)	OpsService:	Register
10	(m)	OpsService:	Event
11	(m)	OpsService::	Activate/Deactivate measurement
12	(m)	OpsService:	Get log data

6.5.6.2.2 Attributes

TCD
This attribute specifies the transaction identification to be used to convey the service and to identify the context of the application relationship as well as the AREP.

State
This attribute indicates the current state of the Message data ASE (FSPM) that is defined in detail in IEC 61158-6-26.

AREP
This attribute indicates information for local identification of the AREP to be used to convey the service and to identify the AR.

Network address
This attribute identifies the network address of the node.

Node number
This attribute identifies the node number of the node.

Common memory space (Area 1, Size 1, Area 2, Size 2)
This attribute identifies the common-memory-area-1 and -2.

Virtual-address-space (Byte block area, Word block area)
This attribute identifies the Virtual-address-space of Byte block and Word block areas.

Node status
This attribute identifies the node status of this node.

List of node management information parameters
This attribute identifies the parameters list of the node management information of the nodes.

Network management information
This attribute identifies the parameters list of the Network management information of the node.

SEQ number list (TX-SEQ, RX-SEQ)
This attribute identifies the sequence number list for the data-transmission and -reception between other nodes.

V_SEQ version (TX-V_SEQ, RX- V_SEQ)
This attribute indicates the version of sequence number for the data-transmission and -reception between other nodes.

Log data memory area
This attribute indicates the memory area information in which the log data is stored.

6.5.6.2.3 Services

Set configuration parameter
The Set configuration parameter request primitive is used to set up the node configuration parameters to the node.

Read node management information parameter
The Read node management information parameter request primitive is used to get the values of the node management information parameters of the node.

Read participating node management information parameter
The Read participating node management information parameter request primitive is used to read out the values of the participating node management information parameters of the nodes participating in the Type 26 fieldbus network.

Read network management information parameter

The Read network management information parameter request primitive is used to read out the network common operating state.

Read message sequence number management information
The Read message sequence number management information request primitive is used to read out the message sequence number management information of the node.

Read node status
The Read node status request primitive is used to read out the node status of the node.

Reset node
The Reset node request primitive is used to reset other nodes.

Set network address
The Set network address request primitive is used to set the network address to the node.

Register
The Register request primitive is used to enroll the AREP to be associated between the services and the UDP/TCP connection.

Event
The Event indication primitive is used to notify the NM-user about certain events or errors in the node.

Activate/Deactivate measurement

The Activate/Deactivate measurement request primitive is used to activate or deactivate the communication load measurement.

Get log data
The Get log data request primitive is used to get the values of the log data on the log data area.

6.5.6.3 Service specification

6.5.6.3.1 Supported services

The services provided by the NM ASE are as follows:

- Set configuration parameter service,
- Read node management information parameter service,
- Read participating node management information parameter service,
- Read network management information parameter service,
- Read message sequence number management information service,
- Read node status service,
- Reset node service,
- Set network address,
- Register,
- Event,
- Activate/Deactivate measurement,
- Get log data.

6.5.6.3.2 Set configuration parameter

6.5.6.3.2.1 Service overview

The Set configuration parameter service is used by the NM-user to set up the node configuration parameters to the node. The configuration parameters specify the operational condition of the node.

6.5.6.3.2.2 Service primitives

The service parameters for this service are shown in Table 37.

Table 37 – Service parameters for Set configuration parameter

Parameter name	Req	Cnf
Argument		
Parameter name	M	
Desired value	M	
Result (+)		S
Result (–)		S
Error Info		U
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Parameter name
This parameter identifies the target configuration parameters to be set.

Desired value
This parameter contains the values of the target configuration parameters to be set.

Table 38 shows the configuration parameters and the values to be set.

Table 38 – Configuration parameters

Parameter name	Description	Value	Data type
Node_number	Node number	1 to 254	USINT
C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT
NDN	Node name by the end user	ASCII character string	char[10]
TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	USINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 µs.	USINT

6.5.6.3.3 Read node management information parameter

6.5.6.3.3.1 Service overview

The Read node management information parameter service is used by the NM-user to read out the values of the node management information parameters of the node.

6.5.6.3.3.2 Service primitives

The service parameters for this service are shown in Table 39.

Table 39 – Read node management information parameter service parameters

Parameter name	Req	Cnf
Argument		
Result (+)		S
Desired value		M
Result (–)		S
Error info		U
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Desired value
This parameter contains the values of the node management information parameters read.

Table 40 shows the node management information parameters and the values.

Table 40 – Node management information parameters

Parameter name	Description	Value	Data type
Node_number	Node number	1 to 254	USINT
C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT

ULS	Upper layer status consists of the operating state and the error information.	– Operating state: "RUN" or "STOP"
		– Error information:
		– "NORMAL" or NOT;
		– "WARNING" or NOT;
		– "ALARM" or NOT;
		– Error_ID
			BitString16

TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	USINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 µs.	USINT
VDN	Vendor code of the node	ASCII character string	char[10]
MSN	Manufacturer model name of the node	ASCII character string	char[10]
NDN	Node name by the end user	ASCII character string	char[10]
P_TYPE	Protocol type	0x80: Fixed value	USINT

LKS	FA link status consists of:
	– "Overlapped_CM_area" state flag;
	– "CM_Set_up_Comp" state flag;
	– "CM_Data_Valid" state flag;
	– "Upper_Layer_inactivity" state flag;
	– "Communication_invalid" state flag;
	– "In_Ring" state flag.
		– "Overlapped_CM_area" or NOT;
		– "CM_Set_up_Comp" or NOT;
		– "CM_Data_Valid" or NOT;
		– "Upper_Layer_inactivity" or NOT;
		– "Communication_invalid" or NOT
		– "In_Ring" or NOT.
			BitString8

Parameter name	Description	Value	Data type

Node_status	Status of this node consists of:
	– "Duplicate_Node_address" state flag;
	– "Overlapped_CM_area" state flag;
	– "TW_timer_expired" state flag;
	– "Wait_for_recv" state flag;
	– "Invalid_Initialize_Para" state flag.
		– "Duplicate_Node" or "NOT";
		– "Overlapped_CM_area";
		– "TW_timer_expired" or "NOT";
		– "Wait_for_recv" or "NOT";
		– "Invalid_Initialize_Para" or "NOT".
			BitString8

6.5.6.3.4 Read participating node management information parameter

6.5.6.3.4.1 Service overview

The Read participating node management information parameter service is used by the NM-user to read out the values of the participating node management information parameters of the target nodes.

6.5.6.3.4.2 Service primitives

The service parameters for this service are shown in Table 41.

Table 41 – Service parameters for Read participating node mgt. information parameter

Parameter name	Req	Cnf
Argument		
Node number	M	
Result (+)		S
Desired value		M
Result (–)		S
Error info		U
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Node number
This parameter indicates the target node number to read out the participating node management information parameters.

Desired value
This parameter contains the values of the participating node management information parameters of the target node.

Table 42 shows the participating node management information parameters and the values.

Table 42 – Participating node management information parameters

Parameter name	Description	Value	Data type
Node_number	Node number	1 to 254	USINT

ULS	Upper layer status consists of the operating state and the error information.	– Operating state: "RUN" or "STOP"
		– Error information:
		– "NORMAL" or NOT;
		– "WARNING" or NOT;
		– "ALARM" or NOT;
		– Error_ID
			BitString16

C_AD1	Data-head-address on common-memory area 1	0 to 0x1FF	UINT
C_SZ1	Data-size on common-memory area 1	0 to 512	UINT
C_AD2	Data-head-address on common-memory area 2	0x0 to 0x1 FFF	UINT
C_SZ2	Data-size on common-memory area 1	0 to 8 192	UINT
RCT	Allowable-refresh-cycle-time	0 to 65 535. The unit is 1 ms.	UINT
TW	Observation time period for the token circulation	1 to 255. The unit is 1 ms.	USINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 µs.	USINT

LKS	FA link status consists of:
	– "Overlapped_CM_area" state flag;
	– "CM_Set_up_Comp" state flag;
	– "CM_Data_Valid" state flag;
	– "Upper_Layer_inactivity" state flag;
	– "Communication_invalid" state flag;
	– "In_Ring" state flag.
		– "Overlapped_CM_area" or NOT;
		– "CM_Set_up_Comp" or NOT;
		– "CM_Data_Valid" or NOT;
		– "Upper_Layer_inactivity" or NOT;
		– "Communication_invalid" or NOT
		– "In_Ring" or NOT.
			BitString8

6.5.6.3.5 Read network management information parameter

6.5.6.3.5.1 Service overview

The Read network management information parameter service is used by the NM-user to read out the network common operating state.

6.5.6.3.5.2 Service primitives

The service parameters for this service are shown in Table 43.

Table 43 – Service parameters for Read network management information parameter

Parameter name	Req	Cnf
Argument		
Result (+)		S
Desired value		M
Result (–)		S
Error info		U
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Desired value
This parameter contains the values of the network management information parameters.

Table 44 shows the network management information parameters and the values.

Table 44 – Network management information parameters

Parameter name	Description	Value	Data type
Token_holding_node_number	Token holding node number	1 to 254	USINT
MFT	Allowable minimum frame interval time	0 to 50. The unit is 100 μs.	USINT
RCT	Allowable-refresh-cycle-time	0 to 65 535. The unit is 1 ms.	UINT
RMT_current	Current value of the Refresh-cycle-measurement-time	0 to 65 535. The unit is 1 ms.	UINT
RMT_Maximum	Maximum value of the Refresh-cycle-measurement-time	0 to 65 535. The unit is 1 ms.	UINT
RMT_Minimum	Minimum value of the Refresh-cycle-measurement-time	0 to 65 535. The unit is 1 ms.	UINT

6.5.6.3.6 Read message sequence number management information

6.5.6.3.6.1 Service overview

The Read message sequence number management information service is used by the NM-user to read out the values of the message sequence number management information of the node.

6.5.6.3.6.2 Service primitives

The service parameters for this service are shown in Table 45.

Table 45 – Service parameters for Read message sequence number management information

Parameter name	Req	Cnf
Argument		
Result (+)		S
Desired value		M
Result (–)		S
Error info		U
NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2.		

Desired value
This parameter contains the values of the Read message sequence number management information of this node.

Table 46 shows the Read message sequence number management information parameters and the values.

Table 46 – Read message sequence number management information parameters

Parameter name	Description	Value	Data type
TX-V_SEQ	Version of sequence number for transmission	Roll-over binary count value of 1 to $2^{32}-1$	UDINT
TX-SEQ for one-to-N	Sequence number for one-to-N broadcast transmission	Roll-over binary count value of 1 to $2^{32}-1$	UDINT
TX-SEQ #n for one-to-one	Sequence number for one-to-one transmission to Node #n	Roll-over binary count value of 1 to $2^{32}-1$	UDINT[256]
RX-V_SEQ #n	Version of sequence number for receiving from Node #n (n = 1 to 254)	Roll-over binary count value of 1 to $2^{32}-1$	UDINT[254]
RX-SEQ #n for one-to-N	Sequence number for receiving one-to-N broadcast from Node #n (n = 1 to 254)	Roll-over binary count value of 1 to $2^{32}-1$	UDINT[254]
RX-SEQ #n for one-to-one	Sequence number for receiving one-to-one type transmission from Node #n (n = 1 to 254)	Roll-over binary count value of 1 to $2^{32}-1$	UDINT[254]

6.5.6.3.7 Read node status

6.5.6.3.7.1 Service overview

The Read node status service is used by the NM-user to read out the node status of the node.

The node status represents the operating status of the upper layer process, the FA-link and the node.

6.5.6.3.7.2 Service primitives

The service parameters for this service are shown in Table 47.

Table 47 – Read node status service parameters

Parameter name	Req	Cnf
Argument		
Result (+)		S
Desired value		M
Result (–)		S
Error info		U

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

Desired value
This parameter contains the values of the Read node status of this node.

Table 48 shows the Read node status variables and the values.

Table 48 – Read node status parameters

Status	Description	Value	Data type

ULS	Upper layer status consists of the operating state and the error information.	– Operating state: "RUN" or "STOP"
		– Error information:
		– "NORMAL" or NOT;
		– "WARNING" or NOT;
		– "ALARM" or NOT;
		– Error_ID
			BitString16

LKS	FA link status consists of:
	– "Overlapped_CM_area" state flag;
	– "CM_Set_up_Comp" state flag;
	– "CM_Data_Valid" state flag;
	– "Upper_Layer_inactivity" state flag;
	– "Communication_invalid" state flag;
	– "In_Ring" state flag.
		– "Overlapped_CM_area" or NOT;
		– "CM_Set_up_Comp" or NOT;
		– "CM_Data_Valid" or NOT;
		– "Upper_Layer_inactivity" or NOT;
		– "Communication_invalid" or NOT
		– "In_Ring" or NOT.
			BitString8

Node_status	Status of this node consists of:
	– "Duplicate_Node_address" state flag;
	– "Overlapped_CM_area" state flag;
	– "TW_timer_expired" state flag;
	– "Wait_for_recv" state flag;
	– "Invalid_Initialize_Para" state flag.
		– "Duplicate_Node" or "NOT";
		– "Overlapped_CM_area" or "NOT";
		– "TW_timer_expired" or "NOT";
		– "Wait_for_recv" or "NOT";
		– "Invalid_Initialize_Para" or "NOT".
			BitString8

Table 49 shows the upper layer operating condition matrix.

Table 49 – Upper layer operating condition matrix

Error Information	Operating state	
	RUN	STOP
NORMAL	The "upper layer process" is in operation, and both of the cyclic-data- and the message-data-transmission are applicable.	The "upper layer process" is halted, and both of the cyclic-data- and the message-data-transmission are applicable.
WARNING	The "upper layer process" is in operation under an error condition, and both of the cyclic-data- and the message-data-transmission are applicable.	The "upper layer process" is halted under the error condition, and both of the cyclic-data- and the message-data-transmission are applicable.
ALARM	Both of the cyclic-data and the message-data-transmission are not applicable.	Both of the cyclic data and message data are not applicable.

6.5.6.3.8 Reset node

6.5.6.3.8.1 Service overview

The Reset node service is used by the NM-user to reset the target node.

6.5.6.3.8.2 Service primitives

The service parameters for this service are shown in Table 50.

Table 50 – Reset node service parameters

Parameter name	Req	Ind	Rsp	Cnf
Argument				
DNA	M		M	
SNA	M	M (=)	M	M
TCD	M	M (=)	M	
AREP	M	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. See 1.2.				

6.5.6.3.9 Set network address

6.5.6.3.9.1 Service overview

The Set network address service is used by the NM-user to set the network address to the node.

6.5.6.3.9.2 Service primitives

The service parameters for this service are shown in Table 51.

Table 51 – Set network address service parameters

Parameter name	Req	Cnf
Argument		
Network address	M	
Address class	M	
Result (+)		S
Result (–)		S
Error info		M

Network address
This parameter specifies the network IP address, the subnet mask for the Type 26 fieldbus network.

Address class
This parameter identifies the subnet address for the local or the router node.

6.5.6.3.10 Register

6.5.6.3.10.1 Service overview

The Register service is used by the NM-user to enroll the AREP to be associated between the services and the communication protocol of UDP or TCP or both of them, to the node.