# IEC 61158-5-18

Edition 2.0 2010-08

# INTERNATIONAL STANDARD

## Industrial communication networks – Fieldbus specifications – Part 5-18: Application layer service definition – Type 18 elements

IEC 61158-5-18:2010(E)

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


### About the IEC

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

### About IEC publications

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

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

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 It also gives information on projects, withdrawn and replaced publications.

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

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

- Electropedia: www.electropedia.org

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

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

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

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

# IEC 61158-5-18

Edition 2.0 2010-08

# INTERNATIONAL STANDARD

## Industrial communication networks – Fieldbus specifications – Part 5-18: Application layer service definition – Type 18 elements

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE W

ICS 25.04.40; 35.100.70; 35.110

ISBN 978-2-88912-111-3

# CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

# INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

## Part 5-18: Application layer service definition – Type 18 elements

## FOREWORD

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

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

International Standard IEC 61158-5-18 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

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

The main changes with respect to the previous edition are listed below:

- Editorial corrections
- Addition of cyclic data segmenting

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

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

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

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

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

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

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

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

# INTRODUCTION

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

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

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

# INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

# Part 5-18: Application layer service definition – Type 18 elements

## 1 Scope

### 1.1 Overview

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

This part of IEC 61158 provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 18 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 part of IEC 61158 defines in an abstract way the externally visible service provided by the Type 18 fieldbus application layer in terms of

a) an abstract model for defining application resources (objects) capable of being manipulated by users via the use of the FAL service,

b) the primitive actions and events of the service;

c) the parameters associated with each primitive action and event, and the form which they take; and

d) the interrelationship between these actions and events, and their valid sequences.

The purpose of this part of IEC 61158 is to define the services provided to

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

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

This part of IEC 61158 specifies the structure and services of the Type 18 IEC fieldbus Application Layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498-1) and the OSI Application Layer Structure (ISO/IEC 9545).

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

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

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

### 1.2 Specifications

The principal objective of this part of IEC 61158 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 specification may be used as the basis for formal Application Programming-Interfaces. Nevertheless, it is not a formal programming interface, and any such interface will need to address implementation issues not covered by this specification, including

a) the sizes and octet ordering of various multi-octet service parameters, and

b) the correlation of paired request and confirm, or indication and response, primitives.

### 1.3 Conformance

This part of IEC 61158 does not specify individual implementations or products, nor do they constrain the implementations of application layer entities within industrial automation systems.

There is no conformance of equipment to this application layer service definition standard. Instead, conformance is achieved through implementation of conforming application layer protocols that fulfill the Type 18 application layer services as defined in this part of IEC 61158.

## 2 Normative references

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

IEC 60559, *Binary floating-point arithmetic for microprocessor systems*

IEC/TR 61158-1:2010[1], *Industrial communication networks – Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series*

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

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

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

_____

1 To be published

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

# 3 Terms, definitions, abbreviations, and conventions

## 3.1 Referenced terms and definitions

### 3.1.1 ISO/IEC 7498-1 terms

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

a) application entity
b) application 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 ISO/IEC 8822 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8822 apply:

a) abstract syntax
b) presentation context

### 3.1.3 ISO/IEC 9545 terms

For the purposes of this document, the following terms as defined 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 ISO/IEC 8824-1 terms

For the purposes of this document, the following terms as defined in ISO/IEC 8824-1 apply:

a) object identifier
b) type

## 3.2 Additional terms and definitions for distributed automation

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

**3.2.1**
**Register X**
register containing bit-oriented cyclic data of type *input data* that is transmitted from a slave to a master

**3.2.2**
**Register Y**
register containing bit-oriented cyclic data of type *output data* that is transmitted from a master to a slave

**3.2.3**
**Register Wr**
register containing word-oriented cyclic data of type *input data* that is transmitted from a slave to a master

**3.2.4**
**Register Ww**
register containing word-oriented cyclic data of type *output data* that is transmitted from a master to a slave

### 3.3 Abbreviations and symbols

| | |
|---|---|
| RX | Register X |
| RY | Register Y |
| RWr | Register Wr |
| RWw | Register Ww |

### 3.4 Conventions

There are no conventions defined specific to Type 18 FAL.

## 4 Concepts

### 4.1 Common concepts

IEC/TR 61158-1:2010, Clause 9 describes the concepts of the application layer service descriptions and the templates used in this document, except as specifically overridden in 4.2.

### 4.2 Type specific concepts

#### 4.2.1 Overview

Described in this communication model specification are the Application Service Elements (ASE) and the Application Process (AP) object class models. The syntax and related encoding of attributes is described by the Type 18 Application layer protocol specification.

The Type 18 AL identifies two types of FAL user, master and slave. For each FAL user type there are two classes of Data Link Entity (DLE), Class 1 and Class 2, corresponding to the Polled and Packed class of the DLE, respectively. See appropriate definitions and specifications in the Type 18 Data link for more information about the Polled and Packed class of DLE. Therefore, there are 4 types of Application Relationship (AR) as shown in Table 1.

**Table 1 – AR types**

| AR class | Symbol | FAL user type | DLE class |
|---|---|---|---|
| Master Class 1 | M1 | Master | Polled |
| Master Class 2 | M2 | Master | Packed |
| Slave Class 1 | S1 | Slave | Polled |
| Slave Class 2 | S2 | Slave | Packed |

An Application Process (AP) object model consists of

- one Device Manager object,
- one Connection Manager object, and
- one or more Process Data objects.

The specific classes of objects included depend upon the type of device and are identified by prefixes that match the symbol of the related AR class.

### 4.2.2 Stations and slots

Each device and its corresponding FAL is identified by a number. This identifying number is named the Station number. The station number is the address used to identify the device and the AR End Point (AREP) associated with the transmission and reception of its process data.

Cyclic process data is further addressed by slot number. One slot is the granularity of the position dependent mapping of the cyclic data fields. A station may occupy more than one slot. The slots belonging to an AREP are identified by the range of slots beginning with the station number and with a length equal to the number of occupied slots as configured by the FAL user.

### 4.2.3 Transmission methods

The Type 18 AL implements a master/slave type architecture. Only a master is able to initiate transmissions. Slave devices respond to transmissions from the master. The access method employed is scanning. A scan cycle is one where the master device transmits data to, and receives data from, all the slaves connected to it. The particulars of the transmission methods are described by the Type 18 Application layer protocol specification.

### 4.2.4 Process data structures

Symbols (RX, RY, RWr, and RWw) are used throughout the Type 18 specifications to refer to types of cyclic data registers which are used to buffer process data for transmission and reception.

In addition to cyclic data, some AR types support acyclic message transmissions, sometimes named *transient data* transmissions by some industry users.

The type of data supported by an FAL is indicated by the level of process data support which is specified with the nomenclature described in Table 2.

**Table 2 – Process data support level**

| Process data support level | Data type supported | Type 18 industry users Alias terminology |
|---|---|---|
| A | bit-oriented i/o data | Remote i/o station |
| B | A + word-oriented i.o data | Remote device station |
| C | B + acyclic messaging | Intelligent device station |

# 5 Data type ASE

## 5.1 General

An overview of the data type ASE and the relationships between data types is provided in IEC/TR 61158-1:2010, 10.1.

## 5.2 Formal definition of data type objects

The template used to describe the data type class in this clause is detailed in IEC/TR 61158-1:2010, 10.2. This includes the specific ASE structure and the definition of its attributes.

## 5.3 FAL defined data types

### 5.3.1 Fixed length types

#### 5.3.1.1 Boolean types

##### 5.3.1.1.1 Boolean

**CLASS:** **Data type**
**ATTRIBUTES:**
1 Data type Numeric Identifier = 1
2 Data type Name = Boolean
3 Format = FIXED LENGTH
4.1 Octet Length = 1

This data type expresses a Boolean data type with the values TRUE and FALSE.

#### 5.3.1.2 Bitstring types

##### 5.3.1.2.1 BitString8

**CLASS:** **Data type**
**ATTRIBUTES:**
1 Data type Numeric Identifier = 22
2 Data type Name = Bitstring8
3 Format = FIXED LENGTH
5.1 Octet Length = 1

##### 5.3.1.2.2 Octet

This data type is the same as Bitstring8.

##### 5.3.1.2.3 BitString16

**CLASS:** **Data type**
**ATTRIBUTES:**
1 Data type Numeric Identifier = 23
2 Data type Name = Bitstring16
3 Format = FIXED LENGTH
5.1 Octet Length = 2

#### 5.3.1.2.4 Word

This data type is the same as Bitstring16.

#### 5.3.1.2.5 BitString32

**CLASS:** **Data type**
**ATTRIBUTES:**

| | | | |
|---|---|---|---|
| 1 | Data type Numeric Identifier | = | 24 |
| 2 | Data type Name | = | Bitstring32 |
| 3 | Format | = | FIXED LENGTH |
| 5.1 | Octet Length | = | 4 |

### 5.3.1.3 Numeric types

#### 5.3.1.3.1 Floating Point types

##### 5.3.1.3.1.1 Float32

**CLASS:** **Data type**
**ATTRIBUTES:**

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

This type has a length of four octets. The format for Float32 is that defined by IEC 60559 as single precision.

##### 5.3.1.3.1.2 float

This data type is the same as Float32.

##### 5.3.1.3.1.3 Float64

**CLASS:** **Data type**
**ATTRIBUTES:**

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

This type has a length of eight octets. The format for Float64 is that defined by IEC 60559 as double precision.

##### 5.3.1.3.1.4 double

This data type is the same as Float64.

#### 5.3.1.3.2 Integer types

##### 5.3.1.3.2.1 Integer8

**CLASS:** **Data type**
**ATTRIBUTES:**

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

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

##### 5.3.1.3.2.2 char

This data type is the same as Integer8.

#### 5.3.1.3.2.3 Integer16

**CLASS:** **Data type**
**ATTRIBUTES:**

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

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

#### 5.3.1.3.2.4 short

This data type is the same as Integer16.

#### 5.3.1.3.2.5 Integer32

**CLASS:** **Data type**
**ATTRIBUTES:**

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

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

#### 5.3.1.3.2.6 long

This data type is the same as Integer32.

### 5.3.1.3.3 Unsigned types

#### 5.3.1.3.3.1 Unsigned8

**CLASS:** **Data type**
**ATTRIBUTES:**

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

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

#### 5.3.1.3.3.2 unsigned char

This data type is the same as Unsigned8.

#### 5.3.1.3.3.3 Unsigned16

**CLASS:** **Data type**
**ATTRIBUTES:**

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

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

#### 5.3.1.3.3.4 unsigned short

This data type is the same as Unsigned16.

#### 5.3.1.3.3.5 Unsigned32

**CLASS:** **Data type**
**ATTRIBUTES:**

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

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

#### 5.3.1.3.3.6 unsigned long

This data type is the same as Unsigned32.

### 5.4 Data type ASE service specification

There are no operational services defined for the type object.

## 6 Communication model specification

### 6.1 General

The Type 18 AL describes a master/slave type architecture with master transmission methods employed to transfer process data between the master device and its connected slave devices.

### 6.2 ASEs

#### 6.2.1 Management ASE

#### 6.2.1.1 Overview

The management ASE manages the FAL components related to the type of device in which the FAL is implemented.

#### 6.2.1.2 Management class specification

#### 6.2.1.2.1 Formal model

**FAL ASE:** **Management ASE**
**CLASS:** **Manager**
**CLASS ID:** **not used**
**PARENT CLASS:** **TOP**
**SERVICES:**

| | | | |
|---|---|---|---|
| 1. | (m) | OpsService: | Get |
| 2. | (o) | OpsService: | Set |
| 3. | (o) | OpsService: | Error |
| 4. | (c) | Constraint: | object class = M1/M2/S1/S2 Connection Manager |
| 4.1. | (m) | OpsService: | Connect |
| 4.2. | (m) | OpsService: | Disconnect |
| 5. | (c) | Constraint: | object class = M1/M2 Connection Manager |
| 5.3. | (m) | OpsService: | Start scan |
| 5.4. | (m) | OpsService: | Stop scan |

#### 6.2.1.2.2 Attributes

There are no attributes defined for this ASE.

#### 6.2.1.2.3 Services

**Get**
This service is used to read attribute values.

**Set**
This service is used to write attribute values.

**Error**
This indication is used to communicate error events.

**Connect**
This service is used to connect to the network.

**Disconnect**
This service is used to disconnect to the network.

**Start scan**
This service is used to initiate the master scanning.

**Stop scan**
This service is used to terminate the master scanning.

### 6.2.1.3 Management ASE service specifications

#### 6.2.1.3.1 Get service

##### 6.2.1.3.1.1 Service overview

This service is used to read attribute values.

##### 6.2.1.3.1.2 Service primitives

The parameters for the Get service are listed in Table 3.

**Table 3 – Get Attributes service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Requested attribute | M | M (=) | | |
| Response code | | | M | M (=) |
| Attribute value | | | 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. | | | | |

**Requested attribute**
This parameter specifies the identity of the attribute whose value is being requested.

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

**Attribute value**
This parameter specifies the value of the requested attribute if successful. The format of the parameter is that of the attribute requested.

##### 6.2.1.3.1.3 Service procedure

The service request results in a returned value or an error code.

#### 6.2.1.3.2 Set service

##### 6.2.1.3.2.1 Service overview

This service is used to write attribute values.

##### 6.2.1.3.2.2 Service primitives

The parameters for the Set service are listed in Table 4.

**Table 4 – Set Attributes service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Requested attribute | M | M (=) | | |
| Attribute value | M | M (=) | | |
| Response code | | | 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. | | | | |

**Requested attribute**
This parameter specifies the identity of the attribute whose value is being set.

**Attribute value**
This parameter specifies the value to which the attribute is to be set. The format of the parameter is that of the attribute being set.

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

##### 6.2.1.3.2.3 Service procedure

The service request results in an attribute value being set or an error code.

#### 6.2.1.3.3 Error indication

##### 6.2.1.3.3.1 Service overview

This indication is used to communicate error events.

##### 6.2.1.3.3.2 Service primitives

The parameters for the Error indication are listed in Table 5.

**Table 5 – Error indication parameters**

| Parameter name | Ind |
|---|---|
| Error code | M |

**Error code**
This parameter specifies the classification of the error.

##### 6.2.1.3.3.3 Service procedure

The FAL user is notified of an error with the accompanying code.

#### 6.2.1.3.4 Connect service

##### 6.2.1.3.4.1 Service overview

This service is used to connect to the network.

##### 6.2.1.3.4.2 Service primitives

The parameters for the Connect service are listed in Table 6.

**Table 6 – Connect service parameters**

| Parameter name | Req | Cnf |
|---|---|---|
| Response code | | M |
| NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2. | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

##### 6.2.1.3.4.3 Service procedure

The Connect service request invokes an appropriate series of DL service requests through the AP protocol in order to establish the specified AR.

For Connection Manger objects belonging to the master class, this service provides the FAL user with connection to, and identification of, one or more slaves to which it has access.

For Connection Manger objects belonging to the slave class, this service provides the FAL user with connection to, and identification of, its associated master and begins the behavior associated with the slave class scanning of process data.

#### 6.2.1.3.5 Disconnect service

##### 6.2.1.3.5.1 Service overview

This service is used to disconnect from the network.

##### 6.2.1.3.5.2 Service primitives

The parameters for the Disconnect service are listed in Table 7.

**Table 7 – Disconnect service parameters**

| Parameter name | Req | Cnf |
|---|---|---|
| Response code | | M |
| NOTE The method by which a confirm primitive is correlated with its corresponding preceding request primitive is a local matter. See 1.2. | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

#### 6.2.1.3.5.3 Service procedure

The Disconnect service request invokes an appropriate series of DL service requests through the AP protocol in order to terminate a previously established AR.

### 6.2.1.3.6 Start scan service

#### 6.2.1.3.6.1 Service overview

This service is used to initiate the master scanning.

#### 6.2.1.3.6.2 Service primitives

The parameters for the Start scan service are listed in Table 8.

**Table 8 – Start scan service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Response code | | | 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. | | | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

#### 6.2.1.3.6.3 Service procedure

The Start scan service request invokes an appropriate series of DL service requests through the AP protocol in order to begin the behavior associated with the scanning of process data.

### 6.2.1.3.7 Stop scan service

#### 6.2.1.3.7.1 Service overview

This service is used to terminate the master scanning.

#### 6.2.1.3.7.2 Service primitives

The parameters for the Stop scan service are listed in Table 9.

**Table 9 – Stop scan service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Response code | | | 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. | | | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

#### 6.2.1.3.7.3 Service procedure

The Stop scan service request invokes an appropriate series of DL service requests through the AP protocol in order to terminate a previously started scan behavior.

#### 6.2.1.4 M1 device manager class specification

##### 6.2.1.4.1 Formal model

The M1 device manager class supports a master type FAL user on a Polled type DL implementation.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Management ASE** |
| **CLASS:** | | | **M1 device manager** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Manager** |
| **ATTRIBUTES:** | | | |
| 1 | (m) | Attribute: | Management information |
| 1.1 | (m) | Attribute: | Transmission speed |
| 1.2 | (m) | Attribute: | Number of occupied stations |
| 1.3 | (m) | Attribute: | Device information |
| 1.3.1 | (m) | Attribute: | Station number |
| 1.3.2 | (m) | Attribute: | Vendor code |
| 1.3.3 | (m) | Attribute: | Model code |
| 1.3.4 | (o) | Attribute: | Software/protocol version |
| 2 | (m) | Attribute: | Connected slaves management information |
| 2.1 | (m) | Attribute: | Slave information 1 |
| 2.1.1 | (m) | Attribute: | Station number |
| 2.1.2 | (m) | Attribute: | Vendor code |
| 2.1.3 | (m) | Attribute: | Model code |
| 2.1.4 | (o) | Attribute: | Software/protocol version |
| 2.1.5 | (m) | Attribute: | Reserved field |
| … | … | … | … |
| 2.n | (m) | Attribute: | Slave information n |
| 2.n.1 | (m) | Attribute: | Station number |
| 2.n.2 | (m) | Attribute: | Vendor code |
| 2.n.3 | (m) | Attribute: | Model code |
| 2.n.4 | (o) | Attribute: | Software/protocol version |
| 2.n.5 | (m) | Attribute: | Reserved field |
| … | … | … | … |
| 2.64 | (m) | Attribute: | Slave information 64 |
| 2.64.1 | (m) | Attribute: | Station number |
| 2.64.2 | (m) | Attribute: | Vendor code |
| 2.64.3 | (m) | Attribute: | Model code |
| 2.64.4 | (o) | Attribute: | Software/protocol version |
| 2.64.5 | (m) | Attribute: | Reserved field |

##### 6.2.1.4.2 Attributes

**Management information**
This attribute is a description of the master's configuration; structure of:

**Transmission speed**
This element specifies the baud rate for the DL configuration.

**Number of occupied stations**
This element specifies the number of stations occupied.

**Station number**
This element specifies the station number for the DL configuration.

**Vendor code**
This element contains a unique code to identify the manufacturer of the device. Vendor code assignment and management is beyond the scope of this specification.

**Model code**
This element specifies the type of device.

**Software/protocol version**
This element specifies the software and protocol versions of the device.

**Connected slaves management information**
This attribute contains a description of the configuration of all connected slaves; array of 64 structures:

**Slave information n**
This element is a description of the configuration of slave n; structure of:

**Station number**
This element specifies the station number for the DL configuration.

**Vendor code**
This element contains a unique code to identify the manufacturer of the device. Vendor code assignment and management is beyond the scope of this specification.

**Model code**
This element specifies the type of device.

**Software/protocol version**
This element specifies the software and protocol versions of the device.

**Reserved field**
This is a reserved field for future definition.

#### 6.2.1.4.3 Services

There are no additional services defined for this class.

### 6.2.1.5 M2 device manager class specification

#### 6.2.1.5.1 Formal model

The M2 device manager class supports a master type FAL user on a Packed type DL implementation.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Management ASE** |
| **CLASS:** | | | **M2 device manager** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Manager** |
| **ATTRIBUTES:** | | | |
| 1 | (m) | Attribute: | Slave station information |
| 1.1 | (m) | Attribute: | Slave station information 1 |
| … | … | … | … |
| 1.n | (m) | Attribute: | Slave station information n |
| … | … | … | … |
| 1.64 | (m) | Attribute: | Slave station information 64 |
| 2 | (m) | Attribute: | Slave station status information |
| 2.1 | (m) | Attribute: | Slave station status information 1 |
| … | … | … | … |
| 2.n | (m) | Attribute: | Slave station status information n |
| … | … | … | … |
| 2.64 | (m) | Attribute: | Slave station status information 64 |

#### 6.2.1.5.2 Attributes

**Slave station information**
This attribute specifies a list of slave station information words collected from each connected slave; array of 64 words:

**Slave station information n**
This element specifies the configuration of the referenced slave device.

**Slave station status information**
This attribute specifies a list of slave station status information fields collected from each connected slave; array of 64, 4-bit elements:

**Slave station status information n**
This element specifies the status of the referenced slave device.

#### 6.2.1.5.3 Services

There are no additional services defined for this class.

### 6.2.1.6 S1 device manager class specification

#### 6.2.1.6.1 Formal model

The S1 device manager class supports a slave type FAL user on a Polled type DL implementation.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Management ASE** |
| **CLASS:** | | | **S1 device manager** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Manager** |
| **ATTRIBUTES:** | | | |
| 1 | (m) | Attribute: | Management information |
| 1.1 | (m) | Attribute: | Station number |
| 1.2 | (m) | Attribute: | Vendor code |
| 1.3 | (m) | Attribute: | Model code |
| 1.4 | (o) | Attribute: | Software/protocol version |
| 1.5 | (m) | Attribute: | Reserved field |

#### 6.2.1.6.2 Attributes

**Management information**
This attribute specifies a description of the slave's configuration; structure of

**Station number**
This element specifies the station number for the DL configuration.

**Vendor code**
This element contains a unique code to identify the manufacturer of the device. Vendor code assignment and management is beyond the scope of this specification.

**Model code**
This element specifies the type of device.

**Software/protocol version**
This element specifies the software and protocol versions of the device.

Reserved field

This is a reserved field for future definition.

#### 6.2.1.6.3 Services

There are no additional services defined for this class.

### 6.2.1.7 S2 device manager class specification

#### 6.2.1.7.1 Formal model

The S2 device manager class supports a slave type AL-user on a Packed type DL implementation.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Management ASE** |
| **CLASS:** | | | **S2 device manager** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Manager** |
| **ATTRIBUTES:** | | | |
| 1 | (m) | Attribute: | Slave station information |

| | | | |
|---|---|---|---|
| 2 | (m) | Attribute: | Slave station status information |

#### 6.2.1.7.2 Attributes

**Slave station information**
This attribute specifies the configuration of the slave device.

**Slave station status information**
This attribute specifies the status of the slave device.

#### 6.2.1.7.3 Services

There are no additional services defined for this class.

## 6.3 ARs

### 6.3.1 Overview

The Type 18 AR uses buffered transport for process data inputs and outputs. Transmission triggering type services are required depending upon the configuration of the instantiated objects.

### 6.3.2 Connection management

#### 6.3.2.1 M1 connection manager class

#### 6.3.2.1.1 Formal model

The M1 connection manager class supports a master type FAL user on a Polled type DL implementation.

The Process data support level for the M1 connection manager class is Level C.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Management ASE** |
| **CLASS:** | | | **M1 connection manager** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Manager** |
| **ATTRIBUTES:** | | | |
| 1 | (m) | Attribute: | Parameter information |
| 1.1 | (m) | Attribute: | Number of connected modules |
| 1.2 | (m) | Attribute: | Number of intelligent devices |
| 1.3 | (m) | Attribute: | Station information |
| 1.4 | (m) | Attribute: | Number of automatic return modules |
| 1.5 | (m) | Attribute: | Number of retries |
| 1.6 | (m) | Attribute: | Delay time setting |
| 1.7 | (o) | Attribute: | Standby master station specification |
| 1.8 | (o) | Attribute: | Operation during master error state |
| 1.9 | (o) | Attribute: | Data link during master error state |
| 1.10 | (o) | Attribute: | Scan mode specification |
| 1.11 | (o) | Attribute: | Reserved station specification |
| 1.12 | (o) | Attribute: | Error invalid station specification |
| 1.13 | (o) | Attribute: | Reserved 1 |
| 1.14 | (o) | Attribute: | Reserved 2 |
| 2 | (m) | Attribute: | Network status information |
| 2.1 | (m) | Attribute: | Master status information |
| 2.2 | (m) | Attribute: | Slave status information |
| 2.2.1 | (m) | Attribute: | Slave status information 1 |
| ... | ... | ... | ... |
| 2.2.n | (m) | Attribute: | Slave status information n |
| ... | ... | ... | ... |
| 2.2.64 | (m) | Attribute: | Slave status information 64 |
| 2.3 | (m) | Attribute: | Master transmitted status field |
| 2.4 | (m) | Attribute: | Slave transmitted status field |
| 2.4.1 | (m) | Attribute: | Slave transmitted status field 1 |
| ... | ... | ... | ... |
| 2.4.n | (m) | Attribute: | Slave transmitted status field n |

| | | | |
|---|---|---|---|
| ... | ... | ... | ... |
| 2.4.64 | (m) | Attribute: | Slave transmitted status field 64 |
| 3 | (m) | Attribute: | Network information |
| 3.1 | (m) | Attribute: | Current link scan time |
| 3.2 | (m) | Attribute: | Minimum link scan time |
| 3.3 | (m) | Attribute: | Maximum link scan time |
| **SERVICES:** | | | |
| 1. | (o) | OpsService: | Verify slave configuration |
| 2. | (c) | Constraint: | Master type = standby |
| 2.1. | (m) | OpsService: | Activate standby |

#### 6.3.2.1.2 Attributes

**Parameter information**
This attribute is a structure of:

**Number of connected modules**
This element specifies the number of slave stations connected to the master.

**Number of intelligent devices**
This element specifies the number of local stations and intelligent device stations connected.

**Station information**
This element specifies the station type and number of occupied slots.

**Number of automatic return modules**
This element specifies the number of slave stations that can be returned to the system within one link scan.

**Number of retries**
This element specifies the retry count for communication errors.

**Delay time setting**
This element specifies the link scan interval.

**Standby master station specification**
This element specifies the station number for the standby master station.

**Operation during master error state**
This element specifies the data link status when an error occurs in the master as STOP or CONTINUE the data link.

**Data link during master error state**
This element specifies the state for process data when an error occurs in communications as HOLD or CLEAR the input/output data.

**Scan mode specification**
This element specifies the link scan to be either free-running (continuous loop) or triggered transmission (loop once).

**Reserved station specification**
This element specifies the reserved station numbers. Although reserved stations are counted as connected stations, a data link error will not occur for reserved stations not connected.

**Error invalid station specification**
This element specifies the error invalid slave station numbers. Slave station errors (at the master) will not occur for error invalid slave stations.

**Reserved 1**
This element is a reserved field for future use by the master.

**Reserved 2**
This element is a reserved field for future use by the connected slaves.

**Network status information**
This attribute is a structure of

**Master status information**
This element specifies the reception and monitoring timer status.

**Slave status information n**
This element specifies the scanning results for slave n.

**Master transmitted status field**
This element contains the transmission results received by each connected slave.

**Slave transmitted status field n**
This element contains the response status for slave n.

**Network information**
This attribute is a structure of

**Current link scan time**
This element specifies the indicated link performance.

**Minimum link scan time**
This element specifies the indicated link performance.

**Maximum link scan time**
This element specifies the indicated link performance.

#### 6.3.2.1.3 Services

##### 6.3.2.1.3.1 Verify slave configuration service

###### 6.3.2.1.3.1.1 Service overview

This service is used to compare the configured parameter information against the actual population of connected slaves.

###### 6.3.2.1.3.1.2 Service primitives

The parameters for the Verify slave configuration service are listed in Table 10.

**Table 10 – M1 Verify slave configuration service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Response code | | | 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. | | | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

###### 6.3.2.1.3.1.3 Service procedure

The Verify slave configuration service request compares the configured Parameter information against the actual population of connected slaves as determined during the Connect procedure and may modify subsequent scan cycles based upon the results.

#### 6.3.2.1.3.2 Activate standby service

##### 6.3.2.1.3.2.1 Service primitives

The parameters for the Activate standby scan service are listed in Table 11.

**Table 11 – Stop scan service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Response code | | | 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. | | | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

##### 6.3.2.1.3.2.2 Service procedure

The Activate standby service request invokes an appropriate series of DL service requests through the AP protocol in order to start the behavior associated with an active master. This procedure skips the actual Connect service methods and instead assumes the role established by the previous master. Prior to invoking this service, the FAL user interchanges its output registers with its input registers such that this master device takes over where the previous master ended its proper operation and slave devices are provided with a seamless transition of process i/o data.

### 6.3.2.2 M2 connection manager class

#### 6.3.2.2.1 Formal model

The M2 connection manager class supports a master type FAL user on a Packed type DL implementation.

The Process data support level for the M2 connection manager class is Level A.

**FAL ASE:** **Management ASE**
**CLASS:** **M2 connection manager**
**CLASS ID:** **not used**
**PARENT CLASS:** **Manager**
**ATTRIBUTES:**

| | | | |
|---|---|---|---|
| 1 | (m) | Attribute: | Parameter information |
| 1.1 | (m) | Attribute: | Transmission speed |
| 1.2 | (m) | Attribute: | Last station number |
| 1.3 | (m) | Attribute: | Point mode setting |
| 1.4 | (m) | Attribute: | Master station i/o points setting |
| 2 | (m) | Attribute: | Network status information |
| 2.1 | (m) | Attribute: | Reception status information |
| 2.2 | (m) | Attribute: | Slave status information |

**SERVICES:**

| | | | |
|---|---|---|---|
| 1. | (o) | OpsService: | Verify slave configuration |

#### 6.3.2.2.2 Attributes

**Parameter information**
This attribute is a structure of

**Transmission speed**
This element specifies the baud rate for the DL.

**Last station number**
This element specifies number of the last station connected to the master.

**Point mode setting**
This element specifies the number of points, corresponding to the bit-width for the i/o data structure.

**Master station i/o point mode setting**
This element specifies the total number of i/o points supported by the master.

**Network status information**
This attribute is a structure of

**Reception status information**
This element contains the reception and monitoring timer status.

**Slave status information**
This element contains the scanning results for slaves.

#### 6.3.2.2.3 Services

##### 6.3.2.2.3.1 Verify slave configuration service

###### 6.3.2.2.3.1.1 Service overview

This service is used to compare the configured parameter information against the actual population of connected slaves.

###### 6.3.2.2.3.1.2 Service Primitives

The parameters for the Verify slave configuration service are listed in Table 12.

**Table 12 – M2 Verify slave configuration service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Response code | | | 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. | | | | |

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

###### 6.3.2.2.3.1.3 Service procedure

The Verify slave configuration service request compares the configured Parameter information against the actual population of connected slaves as determined during the Connect procedure and may modify subsequent scan cycles based upon the results.

#### 6.3.2.3 S1 connection manager class

##### 6.3.2.3.1 Formal model

The S1 connection manager class supports a slave type FAL user on a Polled type DL implementation.

The Process data support level for the S1 connection manager class is specified by the configuration attribute.

**FAL ASE:** **Management ASE**
**CLASS:** **S1 connection manager**
**CLASS ID:** **not used**
**PARENT CLASS:** **Manager**

**ATTRIBUTES:**

| | | | |
|---|---|---|---|
| 1 | (m) | Attribute: | Process data support level |
| 2 | (m) | Attribute: | Network status information |
| 2.1 | (m) | Attribute: | Master transmitted status field |

#### 6.3.2.3.2 Attributes

**Process data support level**
This attribute specifies process data support level for the S1 connection manager instance.

**Network status information**
This attribute is a structure of

**Master transmitted status field**
This element contains the transmission results received by each connected slave.

#### 6.3.2.3.3 Services

There are no additional services defined for this class.

### 6.3.2.4 S2 connection manager class

#### 6.3.2.4.1 Formal model

The S2 connection manager class supports a slave type AL-user on a Packed type DL implementation.

The Process data support level for the S2 connection manager class is Level A.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Management ASE** |
| **CLASS:** | | | **S2 connection manager** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Manager** |
| **ATTRIBUTES:** | | | |
| 1 | (m) | Attribute: | Parameter information |
| 1.1 | (m) | Attribute: | Slave station number |
| 1.2 | (m) | Attribute: | Number of occupied slots |

#### 6.3.2.4.2 Attributes

**Parameter information**
This attribute is a structure of

**Slave station number**
This element specifies the station number for the slave.

**Number of occupied slots**
This element specifies the number of i/o slots occupied by the slave device.

#### 6.3.2.4.3 Services

There are no additional services defined for this class.

## 6.3.3 Process Data AR ASE

### 6.3.3.1 Overview

The Process Data AR ASE manages the flow of process data.

### 6.3.3.2 Process data class specification

#### 6.3.3.2.1 Formal model

| | |
|---|---|
| **FAL ASE:** | **Process Data AR ASE** |
| **CLASS:** | **Process data** |

**CLASS ID:** **not used**
**PARENT CLASS:** **TOP**
**SERVICES:**

| | | | |
|---|---|---|---|
| 1. | (m) | OpsService: | Get |
| 2. | (c) | Constraint: | Required in M1 and M2 devices. Optional otherwise. |
| 2.1 | (o) | OpsService: | Set |
| 3. | (o) | OpsService: | Error |

#### 6.3.3.2.2 Attributes

There are no attributes defined for this class.

#### 6.3.3.2.3 Services

**Get**
This service is used to read attribute values.

**Set**
This service is used to write attribute values.

**Error**
This indication is used to communicate error events.

### 6.3.3.3 Process data AR ASE service specifications

#### 6.3.3.3.1 Get service

##### 6.3.3.3.1.1 Service overview

This service is used to read attribute values.

##### 6.3.3.3.1.2 Service primitives

The parameters for the Get service are listed in Table 13.

**Table 13 – Get attributes service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Requested attribute | M | M (=) | | |
| Response code | | | M | M (=) |
| Attribute value | | | 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.

**Requested attribute**
This parameter specifies the identity of the attribute whose value is being requested.

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

**Attribute value**
This parameter specifies the value of the requested attribute if successful. The format of the parameter is that of the attribute requested.

##### 6.3.3.3.1.3 Service procedure

The service request results in a returned value or an error code.

#### 6.3.3.3.2 Set service

##### 6.3.3.3.2.1 Service overview

This service is used to write attribute values.

##### 6.3.3.3.2.2 Service primitives

The parameters for the Set service are listed in Table 14.

**Table 14 – Set attributes service parameters**

| Parameter name | Req | Ind | Rsp | Cnf |
|---|---|---|---|---|
| Requested attribute | M | M (=) | | |
| Attribute value | M | M (=) | | |
| Response code | | | 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. | | | | |

**Requested attribute**
This parameter specifies the identity of the attribute whose value is being set.

**Attribute value**
This parameter specifies the value to which the attribute is to be set. The format of the parameter is that of the attribute being set.

**Response code**
This parameter specifies the classification of the response as either successful or an error code.

##### 6.3.3.3.2.3 Service procedure

The service request results in an attribute value being set or an error code.

#### 6.3.3.3.3 Error indication

##### 6.3.3.3.3.1 Service overview

This indication is used to communicate error events.

##### 6.3.3.3.3.2 Service primitives

The parameters for the Error indication are listed in Table 15.

**Table 15 – Error indication parameters**

| Parameter name | Ind |
|---|---|
| Error code | M |

**Error code**
This parameter specifies the classification of the error.

##### 6.3.3.3.3.3 Service procedure

The FAL user is notified of an error with the accompanying code.

### 6.3.4 M1 cyclic transmission class specification

#### 6.3.4.1.1 Formal model

The M1 cyclic transmission class supports a master type FAL user in association with an M1 Connection manager.

**FAL ASE:** **Process Data AR ASE**
**CLASS:** **M1 cyclic transmission**
**CLASS ID:** **not used**
**PARENT CLASS:** **Process data**
**ATTRIBUTES:**

| | | | |
|---|---|---|---|
| 1. | (m) | Attribute: | Master status |
| 2. | (m) | Attribute: | Data out |
| 2.1. | (m) | Attribute: | RY data |
| 2.2. | (m) | Attribute: | RWw data |
| 3. | (m) | Attribute: | Data in |
| 3.1. | (m) | Attribute: | Number of modules |
| 3.2. | (m) | Attribute: | Slave input data |
| 3.2.1. | (m) | Attribute: | Data in 1 |
| 3.2.1.1. | (m) | Attribute: | Station number |
| 3.2.1.2. | (m) | Attribute: | Slave status |
| 3.2.1.3. | (m) | Attribute: | RX data 1 |
| 3.2.1.4. | (m) | Attribute: | RWr data 1 |
| … | … | … | … |
| 3.2.n. | (m) | Attribute: | Data in n |
| 3.2.n.1. | (m) | Attribute: | Station number |
| 3.2.n.2. | (m) | Attribute: | Slave status |
| 3.2.n.3. | (m) | Attribute: | RX data n |
| 3.2.n.4. | (m) | Attribute: | RWr data n |
| … | … | … | … |
| 3.2.64. | (m) | Attribute: | Data in 64 |
| 3.2.64.1 | (m) | Attribute: | Station number |
| 3.2.64.2 | (m) | Attribute: | Slave status |
| 3.2.64.3 | (m) | Attribute: | RX data 64 |
| 3.2.64.4 | (m) | Attribute: | RWr data 64 |

**SERVICES:**

| | | | |
|---|---|---|---|
| 1. | (c) | Constraint: | Scan Mode Specification ≠ free running |
| 1.1. | (o) | OpsService: | Trigger transmission |
| 2. | (o) | OpsService: | Data received |

#### 6.3.4.1.2 Attributes

**Master status**
This attribute specifies the master's status field that will be transmitted to all connected slaves.

**Data out**
This attribute is a structure of

**RY data**
This element contains the bit-oriented data outputs for transmission to slaves.

**RWw data**
This element contains the word-oriented data outputs for transmission to slaves.

**Data in**
This attribute is a structure of:

**Number of modules**
This element specifies the number of modules in the following array.

**Slave Input data**
This element is an array containing the input data collected from the connected slave devices.

**Data in n**
Input data structure from slave device n

**Station number**
This element specifies the station number corresponding to the process data structure.

**Slave status**
This element indicates the status of the slave.

**RX data**
This element contains the bit-oriented data inputs as received from slaves.

**RWr data**
This element contains the word-oriented data inputs as received from slaves.

#### 6.3.4.1.3 Services

##### 6.3.4.1.3.1 Trigger transmission

###### 6.3.4.1.3.1.1 Service overview

This service is used to trigger the transmission of process data.

###### 6.3.4.1.3.1.2 Service primitives

The are no parameters for the Trigger transmission service.

###### 6.3.4.1.3.1.3 Service procedure

The Trigger transmission service request invokes an appropriate series of DL service requests through the AP protocol in order to begin the behavior associated with one scan of process data as specified by the configuration.

##### 6.3.4.1.3.2 Data received

###### 6.3.4.1.3.2.1 Service overview

This service is used to communicate process data received from the network.

###### 6.3.4.1.3.2.2 Service primitives

There are no parameters for the Data received indication.

###### 6.3.4.1.3.2.3 Service procedure

The Data received indication is delivered to the FAL user in response to a properly received cyclic data scan. This indication informs the FAL user that fresh process data is available in the Data-in attribute.

### 6.3.5 M2 cyclic transmission class specification

#### 6.3.5.1.1 Formal model

The M2 cyclic transmission class supports a master type FAL user in association with an M2 Connection manager.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Process Data AR ASE** |
| **CLASS:** | | | **M2 cyclic transmission** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Process data** |
| **ATTRIBUTES:** | | | |
| 1. | (m) | Attribute: | RY data |
| 2. | (m) | Attribute: | RX data |
| **SERVICES:** | | | |
| 1. | (o) | OpsService: | Data received |

#### 6.3.5.1.2 Attributes

**RY data**
This attribute specifies the bit-oriented data outputs for transmission to slaves.

**RX data**
This attribute specifies the bit-oriented data inputs as received from slaves.

#### 6.3.5.1.3 Services

##### 6.3.5.1.3.1 Data received

###### 6.3.5.1.3.1.1 Service overview

This service is used to communicate process data received from the network.

###### 6.3.5.1.3.1.2 Service primitives

The are no parameters for the Data received indication.

###### 6.3.5.1.3.1.3 Service procedure

The Data received indication is delivered to the FAL user in response to a properly received cyclic data scan. This indication informs the FAL user that fresh process data is available in the RX-data attribute.

### 6.3.6 S1 cyclic transmission class specification

#### 6.3.6.1.1 Formal model

The S1 cyclic transmission class supports a slave type FAL user in association with an S1 connection manger.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Process Data AR ASE** |
| **CLASS:** | | | **S1 cyclic transmission** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Process data** |
| **ATTRIBUTES:** | | | |
| 1. | (m) | Attribute: | Slave status |
| 2. | (m) | Attribute: | Data out |
| 2.1 | (m) | Attribute: | RY data |
| 2.2 | (c) | Constraint: | Process data support level = B or C |
| 2.2.1 | (m) | Attribute: | RWw data |
| 3. | (m) | Attribute: | Master status |
| 4. | (m) | Attribute: | Data in |
| 4.1 | (m) | Attribute: | RX data |
| 4.2 | (c) | Constraint: | Process data support level = B or C |
| 4.2.1 | (m) | Attribute: | RWr data |
| **SERVICES:** | | | |
| 1. | (o) | OpsService: | Data received |

#### 6.3.6.1.2 Attributes

**Slave status**
This attribute specifies the slave's status field that will be transmitted to the connected master.

**Data out**
This attribute is a structure of

**RY data**
This element contains the bit-oriented data outputs as received from master.

**RWw data**
This element contains the word-oriented data outputs as received from master.

**Master status**
This attribute specifies the master's status field that was received from the connected master.

**Data in**
This attribute is a structure of:

**RX data**
This element contains the bit-oriented data inputs for transmission to master.

**RWr data**
This element contains the word-oriented data inputs for transmission to master.

#### 6.3.6.1.3 Services

##### 6.3.6.1.3.1 Data received

###### 6.3.6.1.3.1.1 Service overview

This service is used to communicate process data received events.

###### 6.3.6.1.3.1.2 Service Primitives

The are no parameters for the Data received indication.

###### 6.3.6.1.3.1.3 Service procedure

The Data received indication is delivered to the FAL user in response to a properly received cyclic data scan. This indication informs the FAL user that fresh process data is available in the Data-in attribute.

### 6.3.7 S2 cyclic transmission class specification

#### 6.3.7.1.1 Formal model

The S2 cyclic transmission class supports a slave type FAL user in association with an S2 connection manger.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Process Data AR ASE** |
| **CLASS:** | | | **S2 cyclic transmission** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Process data** |
| **ATTRIBUTES:** | | | |
| 1. | (m) | Attribute: | RY data |
| 2. | (m) | Attribute: | RX data |
| **SERVICES:** | | | |
| 1. | (o) | OpsService: | Data received |

#### 6.3.7.1.2 Attributes

**RY data**
This attribute specifies the bit-oriented data outputs as received from master.

**RX data**
This attribute specifies the bit-oriented data inputs for transmission to master.

#### 6.3.7.1.3 Services

##### 6.3.7.1.3.1 Data received

###### 6.3.7.1.3.1.1 Service overview

This service is used to communicate process data received events.

#### 6.3.7.1.3.1.2 Service primitives

The are no parameters for the Data received indication.

#### 6.3.7.1.3.1.3 Service procedure

The Data received indication is delivered to the FAL user in response to a properly received cyclic data scan. This indication informs the FAL user that fresh process data is available in the RY-data attribute.

### 6.3.8 Acyclic transmission class specification

#### 6.3.8.1.1 Formal model

The acyclic transmission class supports all ARs related to connection manager process data support level C.

| | | | |
|---|---|---|---|
| **FAL ASE:** | | | **Process Data AR ASE** |
| **CLASS:** | | | **Acyclic transmission** |
| **CLASS ID:** | | | **not used** |
| **PARENT CLASS:** | | | **Process data** |
| **ATTRIBUTES:** | | | |
| 1. | (m) | Attribute: | Message out |
| 2. | (m) | Attribute: | Message in |
| **SERVICES:** | | | |
| 1. | (m) | OpsService: | Send message |
| 2. | (m) | OpsService: | Message received |

#### 6.3.8.1.2 Attributes

**Message out**
This attribute specifies the acyclic data message for transmission.

**Message in**
This attribute specifies the acyclic data message received.

#### 6.3.8.1.3 Services

#### 6.3.8.1.3.1 Send message

#### 6.3.8.1.3.1.1 Service overview

This service is used to send acyclic messages.

#### 6.3.8.1.3.1.2 Service primitives

The are no parameters for the Send message service.

#### 6.3.8.1.3.1.3 Service procedure

The Send message service request invokes an appropriate series of DL service requests through the AP protocol in order to begin the behavior associated with the transmission of an acyclic data transmission.

#### 6.3.8.1.3.2 Message received

#### 6.3.8.1.3.2.1 Service overview

This service is used to communicate acyclic message receipt events.

#### 6.3.8.1.3.2.2 Service primitives

The are no parameters for the Message received indication.

#### 6.3.8.1.3.2.3 Service procedure

The Message received indication is delivered to the FAL user in response to a properly received acyclic data message. This indication informs the FAL user that a fresh acyclic data message is available in the Message-in attribute.

## Bibliography

IEC 61131-1, *Programmable controllers – Part 1: General information*

IEC 6113-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-3-18:2007, *Industrial communication networks – Fieldbus specifications – Part 3-18: Data-link layer service definition – Type 18 elements*

IEC 61158-4-18:2010[2], *Industrial communication networks – Fieldbus specifications – Part 4-18: Data-link layer protocol specification – Type 18 elements*

IEC 61158-6-18:2010[2], *Industrial communication networks – Fieldbus specifications – Part 6-18: Application layer protocol specification – Type 18 elements*

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

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

ISO 8601, *Data elements and interchange formats – Information interchange – Representation of dates and times*

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

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

[2] To be published.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

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