

Industrial communication networks — Fieldbus specifications —

Part 6-20: Application layer protocol specification — Type 20 elements

ICS 25.040.40; 35.100.70

National foreword

This British Standard is the UK implementation of EN 61158-6-20:2008. It is identical with IEC 61158-6-20:2007. Together with all of the other sections of BS EN 61158-6, it supersedes BS EN 61158-6:2004 which is withdrawn.

The UK participation in its preparation was entrusted to Technical Committee AMT/7, Industrial communications — Process measurement and control, including fieldbus.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 June 2008

© BSI 2008

ISBN 978 0 580 61642 6

Amendments/corrigenda issued since publication

Date	Comments

English version

**Industrial communication networks -
Fieldbus specifications -
Part 6-20: Application layer protocol specification -
Type 20 elements
(IEC 61158-6-20:2007)**

Réseaux de communication industriels -
Spécifications des bus de terrain -
Partie 6-20: Spécification des services
des couches d'application -
Éléments de type 20
(CEI 61158-6-20:2007)

Industrielle Kommunikationsnetze -
Feldbusse -
Teil 6-20: Protokollspezifikation
des Application Layer
(Anwendungsschicht) -
Typ 20-Elemente
(IEC 61158-6-20:2007)

This European Standard was approved by CENELEC on 2008-02-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

The text of document 65C/476/FDIS, future edition 1 of IEC 61158-6-20, prepared by SC 65C, Industrial networks, of IEC TC 65, Industrial-process measurement, control and automation, was submitted to the IEC-CENELEC parallel vote and was approved by CENELEC as EN 61158-6-20 on 2008-02-01.

This and the other parts of the EN 61158-6 series supersede EN 61158-6:2004.

With respect to EN 61158-6:2004 the following changes were made:

- deletion of Type 6 fieldbus for lack of market relevance;
- addition of new fieldbus types;
- partition into multiple parts numbered 6-2, 6-3, ...6-20.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2008-11-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2011-02-01

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

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 61158-6-20:2007 was approved by CENELEC as a European Standard without any modification.

CONTENTS

INTRODUCTION	6
1 Scope	7
1.1 General	7
1.2 Specifications	7
1.3 Conformance	7
2 Normative references	8
3 Terms, definitions, symbols, abbreviations and conventions	9
3.1 Terms and definitions from other ISO/IEC standards	9
3.2 IEC/TR 61158-1 terms	9
3.3 Type 20 fieldbus application-layer specific definitions	12
3.4 Abbreviations and symbols	14
3.5 Conventions	15
3.6 Conventions used in state machines	15
4 Abstract syntax	16
5 Transfer syntax	16
5.1 General	16
5.2 Common APDU structure	17
5.3 Service-specific APDU structures	19
5.4 Data coding rules	34
6 Structure of FAL protocol state machines	39
7 AP-context state machines	40
8 FAL service protocol machine (FSPM)	40
8.1 General	40
8.2 FSPM state tables	41
8.3 Functions used by FSPM	46
8.4 Parameters of FSPM/ARPM primitives	46
9 Application relationship protocol machines (ARPMs)	47
9.1 AREP mapping to data link layer	47
9.2 Application relationship protocol machines (ARPMs)	48
9.3 AREP state machine primitive definitions	50
9.4 AREP state machine functions	50
10 DLL mapping protocol machine (DMPM)	50
10.1 DMPM states	50
10.2 DMPM state machines	51
10.3 Primitives exchanged between data link layer and DMPM	51
10.4 Functions used by DMPM	52
Bibliography	53
Annex ZA (normative) Normative references to international publications with their corresponding European publications	54
Figure 1 – APDU format	17
Figure 2 – Normal response from slave to master	17
Figure 3 – Command error response from slave to master	18
Figure 4 – Communication error response from slave to master	19
Figure 5 – Coding without identification	34
Figure 6 – Coding of Integer type data	34

Figure 7 – Coding of Integer16 type data	35
Figure 8 – Coding of Unsigned type data	35
Figure 9 – Coding of Unsigned16 type data.....	35
Figure 10 – Coding of single precision Floating Point type data	35
Figure 11 – Coding of double precision Floating Point type data	36
Figure 12 – Coding of Date type data.....	36
Figure 13 – Relationships among protocol machines and adjacent layers	40
Figure 14 – State transition diagram of FSPM.....	41
Figure 15 – State transition diagram of the client ARPM	48
Figure 16 – State transition diagram of the server ARPM	49
Figure 17 – State transition diagram of DMPM	51
Table 1 – Conventions used for state machines	15
Table 2 – Response code values	18
Table 3 – Device status values	18
Table 4 – Response code values	19
Table 5 – Communication error codes.....	19
Table 6 – Identify request APDU	20
Table 7 – Identify response value field	21
Table 8 – Identify command specific response codes.....	21
Table 9 – Read primary variable response value field	22
Table 10 – Read primary variable command specific response codes	22
Table 11 – Read loop current and percent of range value field.....	22
Table 12 – Read loop current and percent of range command specific response codes	23
Table 13 – Read dynamic variables and loop current value field	23
Table 14 – Read dynamic variables and loop current command specific response codes.....	23
Table 15 – Write polling address value field	24
Table 16 – Loop current mode codes	24
Table 17 – Write polling address command specific response codes.....	24
Table 18 – Read loop configuration value field.....	25
Table 19 – Read loop configuration command specific response codes	25
Table 20 – Read dynamic variable families classifications value field.....	25
Table 21 – Read dynamic variable families classifications command specific response codes.....	26
Table 22 – Read device variables with status request value field	26
Table 23 – Read device variables with status command specific response codes.....	26
Table 24 – Read device variables with status value field.....	27
Table 25 – Variable status values	27
Table 26 – Read message response value field	28
Table 27 – Read message command specific response codes	28
Table 28 – Read tag, descriptor, date response value field	29
Table 29 – Read tag, descriptor, date command specific response codes	29

Table 30 – Read primary variable transducer information response value field.....	29
Table 31 – Read primary variable transducer information command specific response codes.....	30
Table 32 – Read device information response value field.....	30
Table 33 – Read device information command specific response codes.....	31
Table 34 – Read final assembly number response value field	31
Table 35 – Read final assembly number command specific response codes	31
Table 36 – Write message value field.....	31
Table 37 – Write message command specific response codes	32
Table 38 – Write tag, descriptor, date value field	32
Table 39 – Write tag, descriptor, date command specific response codes	32
Table 40 – Write final assembly number value field.....	33
Table 41 – Write final assembly number command specific response codes	33
Table 42 – Read long tag response value field.....	33
Table 43 – Read long tag command-specific response codes	33
Table 44 – Write long tag value field	34
Table 45 – Write long tag command specific Response codes	34
Table 46 – Coding for Date type	36
Table 47 – Coding for one octet Enumerated Type.....	37
Table 48 – One octet bit field	38
Table 49 – Packed ASCII character set.....	38
Table 50 – Acceptable subset of ISO Latin-1 characters	39
Table 51 – FSPM state table – client transactions	41
Table 52 – FSPM state table – server transactions	45
Table 53 – Function Command ()	46
Table 54 – Function CommErr ()	46
Table 55 – Function CommandErr ()	46
Table 56 – Function Resp ().....	46
Table 57 – Function Device ()	46
Table 58 – Parameters used with primitives exchanged between FSPM and ARPM	46
Table 59 – Client ARPM states	48
Table 60 – Client ARPM state table	49
Table 61 – Server ARPM states	49
Table 62 – Server ARPM state table	49
Table 63 – Primitives issued from ARPM to DMPM	50
Table 64 – Primitives issued by DMPM to ARPM.....	50
Table 65 – Parameters used with primitives exchanged between ARPM and DMPM	50
Table 66 – DMPM state descriptions	51
Table 67 – DMPM state table – Client transactions	51
Table 68 – DMPM state table – Server transactions	51
Table 69 – Primitives exchanged between data-link layer and DMPM	52

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 protocol provides the application service by making use of the services available from the data-link or other immediately lower layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer application entities (AEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- as a guide for implementors and designers;
- for use in the testing and procurement of equipment;
- as part of an agreement for the admittance of systems into the open systems environment;
- as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 6-20: Application layer protocol specification – Type 20 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 standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 20 fieldbus. The term “time-critical” is used to represent the presence of a time-window, within which one or more specified actions are required to be completed with some defined level of certainty. Failure to complete specified actions within the time window risks failure of the applications requesting the actions, with attendant risk to equipment, plant and possibly human life.

This standard defines in an abstract way the externally visible behavior provided by the Type 20 of the fieldbus Application Layer in terms of

- a) the abstract syntax defining the application layer protocol data units conveyed between communicating application entities,
- b) the transfer syntax defining the application layer protocol data units conveyed between communicating application entities,
- c) the application context state machine defining the application service behavior visible between communicating application entities; and
- d) the application relationship state machines defining the communication behavior visible between communicating application entities; and.

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

- 1) the wire-representation of the service primitives defined in IEC 61158-5-20, and
- 2) the externally visible behavior associated with their transfer.

This standard specifies the protocol of the Type 20 IEC fieldbus application layer, in conformance with the OSI Basic Reference Model (ISO/IEC 7498) and the OSI Application Layer Structure (ISO/IEC 9545).

1.2 Specifications

The principal objective of this standard is to specify the syntax and behavior of the application layer protocol that conveys the application layer services defined in IEC 61158-5-20.

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 protocols standardized in IEC 61158-6.

1.3 Conformance

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

Conformance is achieved through implementation of this application layer protocol specification.

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 61158-5-20, *Industrial communication networks – Fieldbus specifications – Part 5-20: Application layer service definition – Type 20 elements*

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

ISO/IEC 8824, *Information technology – Open Systems Interconnection – Specification of Abstract Syntax Notation One (ASN.1)*

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

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

3 Terms, definitions, symbols, abbreviations and conventions

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

3.1 Terms and definitions from other ISO/IEC standards

3.1.1 Terms and definitions from ISO/IEC 7498-1

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

3.1.2 Terms and definitions from ISO/IEC 9545

- 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

3.1.3 Terms and definitions from ISO/IEC 8824

- a) object identifier
- b) type
- c) value
- d) simple type
- e) structured type
- f) component type
- g) tag
- i) true
- j) false
- k) integer type
- m) octet string type
- n) null type
- o) sequence type
- p) sequence of type
- q) choice type
- r) tagged type
- s) any type
- t) module
- u) production

3.1.4 Terms and definitions from ISO/IEC 8825

- a) encoding (of a data value)
- b) data value
- c) identifier octets (the singular form is used in this standard)
- d) length octet(s) (both singular and plural forms are used in this standard)
- e) contents octets

3.2 IEC/TR 61158-1 terms

The following IEC/TR 61158-1 terms apply.

3.2.1 application

function or data structure for which data is consumed or produced

3.2.2 application layer interoperability

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

3.2.3 application object

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

NOTE: Multiple types of application object classes may be defined

3.2.4 application process

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

3.2.5 application process identifier

identifier that distinguishes among multiple application processes used in a device

3.2.6 application process object

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

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

3.2.7 application process object class

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

3.2.8 application relationship

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

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

3.2.9 application relationship application service element

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

3.2.10 application relationship endpoint

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

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

3.2.11 attribute

description of an externally visible characteristic or feature of an object

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

3.2.12 behaviour

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

3.2.13 class

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

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

3.2.14 class attributes

attribute that is shared by all objects within the same class

3.2.15 class code

unique identifier assigned to each object class

3.2.16 class specific service

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

NOTE A class specific object is unique to the object class which defines it.

3.2.17 client

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

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

3.2.18 conveyance path

unidirectional flow of APDUs across an application relationship

3.2.19 cyclic

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

3.2.20 dedicated AR

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

3.2.21 device

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

3.2.22 device profile

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

3.2.23 endpoint

one of the communicating entities involved in a connection

3.2.24 error

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

3.2.25 error code

identification of a specific type of error within an error class

3.2.26 management information

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

NOTE Managing includes functions such as controlling, monitoring, and diagnosing.

3.2.27 network

series of nodes connected by some type of communication medium

NOTE The connection paths between any pair of nodes can include repeaters, routers and gateways.

3.2.28 pre-defined AR endpoint

AR endpoint that is defined locally within a device without use of the create service

NOTE Pre-defined ARs that are not pre-established are established before being used.

3.2.29 pre-established AR endpoint

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

3.2.30 server

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

3.2.31 service

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

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

3.3 Type 20 fieldbus application-layer specific definitions

There are additional terms defined for this part.

3.3.1 analog channel

continuously varying electrical signal connecting a field device to the remainder of the data acquisition or control system. Some field devices support multiple analog channels (input or output)

NOTE Each analog channel transmits a single dynamic variable to or from the field device.

3.3.2 broadcast address

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

NOTE The broadcast address is five octet long and has all zeros as the value.

3.3.3 busy

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

3.3.4 device ID

a serial number for a device

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

3.3.5 device type

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

NOTE The value of this attribute is assigned by the manufacturer. Its value specifies the set of commands and data objects supported by the device. The manufacturer is required to assigned unique value to each type of the device.

3.3.6 device variable

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

3.3.7 dynamic variable

connection between a process and an analog channel

NOTE A device may contain primary, secondary, tertiary, and quaternary variables. These are collectively called the dynamic variables.

3.3.8 long tag

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

3.3.9 loop current

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

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

3.3.10 manufacturer ID

string identifying the manufacturer that produced a device

NOTE A manufacturer is required to use the value assigned to it and is not permitted to use the value assigned to another manufacturer.

3.3.11 master

device that initiates communication activity by sending request PDUs to other devices

3.3.12 polling address

integer used to identify a device

NOTE The polling address is used to construct a one-octet address.

3.3.13 slave

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

3.3.14 tag

8-character ASCII string used to identify a field device

3.3.15 unique address

five-octet address of a device that uniquely identifies the device among all other devices that support this standard, formed by concatenating a manufacturer ID, a device type and a device ID

3.4 Abbreviations and symbols

AE	Application entity
AL	Application layer
ALP	Application layer protocol
APO	Application object
AP	Application process
APDU	Application protocol data unit
API	Application process Identifier
AR	Application relationship
AREP	Application relationship endpoint
ASCII	American Standard Code for Information Interchange
ASE	Application Service Element
Cnf	Confirmation
DL-	(as a prefix) data-link-
DLC	Data-link connection
DLCEP	Data-link connection endpoint
DLL	Data-link layer
DLM	Data-link-management
DLSAP	Data-link service access point
DLSDU	DL-service-data-unit
FAL	Fieldbus application layer
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
MSB	Most significant byte

LSB	Least significant byte
OSI	Open Systems Interconnect
Req	Request
Rsp	Response
SAP	Service access point
SDU	Service data unit
VFD	Virtual field device

3.5 Conventions

3.5.1 General concept

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: its class definitions, its services, and its protocol specification. The first two are contained in IEC 61158-5. The protocol specification for each of the ASEs is defined in this standard.

The class definitions define the attributes of the classes supported by each ASE. The attributes are accessible from instances of the class using the Management ASE services specified in IEC 61158-5 standard. The service specification defines the services that are provided by the ASE.

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

3.5.2 Conventions for class definitions

The data-link layer mapping definitions are described using templates. Each template consists of a list of attributes for the class. The general form of the template is defined in IEC 61158-5-5.

3.5.3 Abstract syntax conventions

When the "optionalParametersMap" parameter is used, a bit number which corresponds to each OPTIONAL or DEFAULT production is given as a comment.

3.6 Conventions used in state machines

The state machines are described in Table 1.

Table 1 – Conventions used for state machines

#	Current state	Event / condition => action	Next state
Name of this transition	The current state to which this state transition applies	Events or conditions that trigger this state transaction. => The actions that are taken when the above events or conditions are met. The actions are always indented below events or conditions	The next state after the actions in this transition is taken

The conventions used in the state machines are as follows:

:= Value of an item on the left is replaced by value of an item on the right. If an item on the right is a parameter, it comes from the primitive shown as an input event.

xxx A parameter name.

Example:

Identifier := reason
means value of a 'reason' parameter is assigned to a parameter called 'Identifier.'

"xxx" Indicates fixed value.

Example:

Identifier := "abc"
means value "abc" is assigned to a parameter named 'Identifier.'

= A logical condition to indicate an item on the left is equal to an item on the right.

< A logical condition to indicate an item on the left is less than the item on the right.

> A logical condition to indicate an item on the left is greater than the item on the right.

<> A logical condition to indicate an item on the left is not equal to an item on the right.

&& Logical "AND"

|| Logical "OR"

This construct allows the execution of a sequence of actions in a loop within one transition. The loop is executed for all values from start_value to end_value.

Example:

```
for (Identifier := start_value to end_value)
  actions
endfor
```

This construct allows the execution of alternative actions depending on some condition (which might be the value of some identifier or the outcome of a previous action) within one transition.

Example:

```
If (condition)
  actions
else
  actions
endif
```

Readers are strongly recommended to refer to the subclauses for the AREP attribute definitions, the local functions, and the FAL-PDU definitions to understand protocol machines. It is assumed that readers have sufficient knowledge of these definitions, and they are used without further explanations.

4 Abstract syntax

The abstract syntax of APDUs is combined with their transfer syntax and is specified in Clause 5.

5 Transfer syntax

5.1 General

The sending Application Layer prepares an APDU to transfer it to the receiving Application Layer. It uses the parameters from the service primitives to do so. There are several formats of the APDU:

- Request APDU from Master to Slave device,
- Normal response from Slave to Master device,
- Command error response from Slave to Master device, and
- Communication error response from Slave to Master device.