IEC 61158-5-3

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements

IEC 61158-5-3:2007(E)

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

About the IEC

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

About IEC publications

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

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 It also gives information on projects, withdrawn and replaced publications.

- IEC Just Published: www.iec.ch/online_news/justpub
Stay up to date on all new IEC publications. Just Published details twice a month all new publications released. Available on-line and also by email.

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

- Customer Service Centre: www.iec.ch/webstore/custserv
If you wish to give us your feedback on this publication or need further assistance, please visit the Customer Service Centre FAQ or contact us:

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

IEC 61158-5-3

Edition 1.0 2007-12

INTERNATIONAL STANDARD

Industrial communication networks – Fieldbus specifications – Part 5-3: Application layer service definition – Type 3 elements

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE XH

ICS 25.040.40; 35.100.70

ISBN 2-8318-9445-X

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 5-3: Application Layer service definition – Type 3 elements

FOREWORD

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

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

International Standard IEC 61158-5-3 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This first edition and its companion parts of the IEC 61158-5 subseries cancel and replace IEC 61158-5:2003. This edition of this part constitutes an editorial revision.

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

a) deletion of the former Type 6 fieldbus for lack of market relevance;
b) addition of new types of fieldbuses;

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

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

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

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

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

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

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

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

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

INTRODUCTION

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

The application service is provided by the application protocol making use of the services available from the data-link or other immediately lower layer. This standard defines the application service characteristics that fieldbus applications and/or system management 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 5-3: Application Layer service definition – Type 3 elements

1 Scope

1.1 General

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

This sub-part contains material specific to Type 3 fieldbus.

1.2 Overview

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

This standard provides common elements for basic time-critical and non-time-critical messaging communications between application programs in an automation environment and material specific to Type 3 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 define in an abstract way the externally visible service provided by the different Types of fieldbus Application Layer in terms of

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

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

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

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

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

object (APO) classes. One of the FAL ASEs is a management ASE that provides a common set of services for the management of the instances of FAL classes.

Although these services specify, from the perspective of applications, how request and responses are issued and delivered, they do not include a specification of what the requesting and responding applications are to do with them. That is, the behavioral aspects of the applications are not specified; only a definition of what requests and responses they can send/receive is specified. This permits greater flexibility to the FAL users in standardizing such object behavior. In addition to these services, some supporting services are also defined in this standard to provide access to the FAL to control certain aspects of its operation.

1.3 Specifications

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

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

1.4 Conformance

This standard do 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 any given Type of 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 61158-3-3, *Industrial communication networks – Fieldbus specifications - Part 3-3: Data-link layer service definition – Type 3 elements*

IEC 61158-4-3, *Industrial communication networks – Fieldbus specifications - Part 4-3: Data-link layer protocol specification – Type 3 elements*

IEC 61158-6-3, *Industrial communication networks – Fieldbus specifications - Part 6-3: Application layer protocol specification – Type 3 elements*

IEC 61158-5-10, *Industrial communication networks – Fieldbus specifications - Part 5-10: Application layer service definition – Type 10 elements*

IEC 61158-6-10, *Industrial communication networks – Fieldbus specifications - Part 6-10: Application layer protocol specification – Type 10 elements*

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

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

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

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

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

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

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

3 Terms, definitions, abbreviations, symbols 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 terms

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

a) object identifier
b) type

3.1.5 Fieldbus Data Link Layer terms

For the purposes of this document, the following terms as defined in IEC 61158-3-3 and IEC 61158-4-3 apply.

a) DL-Time
b) DL-Scheduling-policy
c) DLCEP
d) DLC
e) DL-connection-oriented mode
f) DLPDU
g) DLSDU
h) DLSAP
i) fixed tag
j) generic tag
k) link
l) MAC ID
m) network address
n) node address
o) node
p) tag
q) scheduled
r) unscheduled.

3.2 Fieldbus Application Layer type 3 – specific terms and definitions

For the purposes of this part of IEC 61158, the following terms and definitions apply.

3.2.1
access protection
limitation of the usage of an application object to one client

3.2.2
address assignment table
mapping of the client's internal I/O-Data object storage to the decentralized input and output data objects

3.2.3
allocate
take a resource from a common area and assign that resource for the exclusive use of a specific entity

3.2.4
application
function or data structure for which data is consumed or produced

3.2.5
application layer interoperability
capability of application entities to perform coordinated and cooperative operations using the services of the FAL

3.2.6
application objects
multiple object classes that manage and provide a run time exchange of messages across the network and within the network device

3.2.7
application process identifier
distinguishes multiple application processes used in a device

3.2.8
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.

3.2.9
application process object class
a class of application process objects defined in terms of the set of their network-accessible attributes and services

3.2.10
application relationship
cooperative association between two or more application-entity-invocations for the purpose of exchange of information and coordination of their joint operation. This relationship is activated either by the exchange of application-protocol-data-units or as a result of preconfiguration activities

3.2.11
application relationship application service element
application-service-element that provides the exclusive means for establishing and terminating all application relationships

3.2.12
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.

3.2.13
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 behavior of an object. Attributes are divided into class attributes and instance attributes.

3.2.14
behavior
indication of how an object responds to particular events

3.2.15
channel
single physical or logical link of an input or output application object of a server to the process

3.2.16
channel related diagnosis
information concerning a specific element of an input or output application object, provided for maintenance purposes

EXAMPLE: validity of data

3.2.17
class
a set of objects, all of which represent the same kind of system component

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

3.2.18
class attributes
attribute that is shared by all objects within the same class

3.2.19
class code
unique identifier assigned to each object class

3.2.20
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.

3.2.21
client
<object view> object which uses the services of another (server) object to perform a task

3.2.22
client
<communication view> initiator of a message to which a server reacts

3.2.23
configuration check
comparison of the expected I/O-Data object structuring of the client with the real I/O-Data object structuring to the server in the start-up phase

3.2.24
configuration fault
an unacceptable difference between the expected I/O-Data object structuring and the real I/O-Data object structuring, as detected by the server

3.2.25
configuration identifier
representation of a portion of I/O Data of a single input- and/or output-module of a server

3.2.26
connection
logical binding between application objects that may be within the same or different devices

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

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

3.2.27
consume
act of receiving data from a producer

3.2.28
consumer
node or sink that is receiving data from a producer

3.2.29
consuming application
application that consumes data

3.2.30
cyclic
repetitive in a regular manner

3.2.31
data consistency
means for coherent transmission and access of the input- or output-data object between and within client and server

3.2.32
default DL-address
value 126 as an initial value for DL-address, which has to be changed (e.g. by assignment of an DL-address via the fieldbus) before operation with a DP-master (class 1)

3.2.33
device
physical hardware connected to the link

NOTE A device may contain more than one node.

3.2.34
diagnosis information
all data available at the server for maintenance purposes

3.2.35
diagnosis information collection
system diagnosis information that is assembled at the client side

3.2.36
DP-master (class 1)
a controlling device which controls several DP-slaves (field devices)

NOTE This is usually a programmable controller or a distributed control system.

3.2.37
DP-master (class 2)
controlling device which manages configuration data (parameter sets) and diagnosis data of a DP-master (Class 1), and that additionally can perform all communication capabilities of a DP-master (Class 1)

3.2.38
DP-slave
field device that can be assigned to one DP-master (Class 1) as a provider for cyclic I/O data exchange; in addition acyclic functions and alarms could be provided

3.2.39
endpoint
one of the communicating entities involved in a connection

3.2.40
engineering
abstract term that characterizes the client application or device responsible for configuring an automation system via interconnecting data items

3.2.41
error
discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.2.42
error class
general grouping for related error definitions and corresponding error codes

3.2.43
error code
identification of a specific type of error within an error class

3.2.44
event
an instance of a change of conditions

3.2.45
FAL subnet
subnetworks composed of one or more data link segments, identified by a subset of the network address

NOTE FAL subnets are permitted to contain bridges but not routers.

3.2.46
freeze
function at the DP-slaves for simultaneous data transfer between the input data object and the process

3.2.47
group
<general> a general term for a collection of objects. Specific uses:

3.2.48
group
<addressing> when describing an address, an address that identifies more than one entity

3.2.49
group
<Type 3> set of DP-slaves which perform a Freeze or Sync function

3.2.50
interface
<general> shared boundary between two functional units, defined by functional characteristics, signal characteristics, or other characteristics as appropriate

3.2.51
interface
<FAL> collection of FAL class attributes and services that represents a specific view on the FAL class

3.2.52
interface pointer
key attribute that unambiguously addresses an object interface instance

3.2.53
invocation
act of using a service or other resource of an application process

NOTE Each invocation represents a separate thread of control that may be described by its context. Once the service completes, or use of the resource is released, the invocation ceases to exist. For service invocations, a service that has been initiated but not yet completed is referred to as an outstanding service invocation. Also for service invocations, an Invoke ID may be used to unambiguously identify the service invocation and differentiate it from other outstanding service invocations.

3.2.54
I/O data
object designated to be transferred cyclically for the purpose of processing

3.2.55
ident number
DP-master (Class 1) or DP-slave device type

3.2.56
identifier related diagnosis
information dedicated to modules for maintenance purpose

3.2.57
index
address of an object within an application process

3.2.58
instance
the actual physical occurrence of an object within a class that identifies one of many objects within the same object class

EXAMPLE California is an instance of the object class state.

NOTE The terms object, instance, and object instance are used to refer to a specific instance.

3.2.59
instantiated
object that has been created in a device

3.2.60
manufacturer ID
identification of each product manufacturer by a unique number

3.2.61
master parameter set
the configuration and parameterization data of all DP-slaves that are assigned to the corresponding DP-master and the bus parameters

3.2.62
member
piece of an attribute that is structured as an element of an array

3.2.63
message router
object within a node that distributes messaging requests to appropriate application objects

3.2.64
method
a synonym for an operational service which is provided by the server ASE and invoked by a client

3.2.65
module
<general> hardware or logical component of a physical device

3.2.66
module
<Type 3> addressable unit inside the DP-slave

3.2.67
network
a set of nodes connected by some type of communication medium, including any intervening repeaters, bridges, routers and lower-layer gateways

3.2.68
object
abstract representation of a particular component within a device, usually a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behavior

3.2.69
physical device
<general> an automation or other network device

3.2.70
process data
object(s) which are already pre-processed and transferred acyclically for the purpose of information or further processing

3.2.71
produce
act of sending data to be received by a consumer

3.2.72
producer
node that is responsible for sending data

3.2.73
provider
source of a data connection

3.2.74
publisher
role of an AR endpoint that transmits APDUs onto the fieldbus for consumption by one or more subscribers

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

3.2.75
real configuration
input and output data structure of the DP-slave, including definition of data consistency

3.2.76
resource
a processing or information capability of a subsystem

3.2.77
server
< communication view> role of an AREP in which it returns a confirmed service response APDU to the client that initiated the request

3.2.78
server
< object view> object which provides services to another (client) object

3.2.79
service
operation or function than an object and/or object class performs upon request from another object and/or object class

3.2.80
slot
address of a module within a DP-slave

3.2.81
subscriber
role of an AREP in which it receives APDUs produced by a publisher

3.2.82
sync
function at the DP-slaves for simultaneous data transfer between the output data object and the process

3.2.83
target
end-node to which a connection is established

3.2.84
unconnected service
messaging service which does not rely on the set up of a connection between devices before allowing information exchanges

3.3 Abbreviations and symbols

AE	Application Entity
AL	Application Layer
ALME	Application Layer Management Entity
ALP	Application Layer Protocol
APO	Application Object
AP	Application Process
APDU	Application Protocol Data Unit
API	Application Process Identifier
AR	Application Relationship
AREP	Application Relationship End Point
ASE	Application Service Element
Cnf	Confirmation
CR	Communication Relationship
CREP	Communication Relationship End Point
DL-	(as a prefix) data-link-
DLL	Data-link layer
DLM	Data-link-management
DLSAP	Data-link Service Access Point
DLSDU	DL-service-data-unit
DP	Decentralised Peripherals
FAL	Fieldbus Application Layer
ID	Identifier
IEC	International Electrotechnical Commission
Ind	Indication
ISO	International Organization for Standardization
LME	Layer Management Entity
OSI	Open Systems Interconnect
PDev	Physical Device
PDU	Protocol Data Unit
PL	Physical Layer
QoS	Quality of Service
Req	Request
Rsp	Response
SAP	Service Access Point

3.4 Conventions

3.4.1 Overview

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

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

3.4.2 General conventions

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

3.4.3 Conventions for class definitions

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

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

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

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

(3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2 048 are allocated for identifying user defined classes.

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

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

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

a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.

b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.

c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify

i) fields of a structured attribute (4.1, 4.2, 4.3),

ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2).

iii) the selection fields of a choice type attribute (6.1 and 6.2).

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

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

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

c) The label "MgtService" designates an management service (2).

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

3.4.4 Conventions for service definitions

3.4.4.1 General

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

3.4.4.2 Service parameters

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

The service specifications of this standard uses a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

1) Parameter name,
2) request primitive,
3) indication primitive,
4) response primitive, and
5) confirm primitive.

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column:

M parameter is mandatory for the primitive

U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.

C parameter is conditional upon other parameters or upon the environment of the service user.

— (blank) parameter is never present.

S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

a) a parameter-specific constraint:
"(=)" indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.

b) an indication that some note applies to the entry:
"(n)" indicates that the following note "n" contains additional information pertaining to the parameter and its use.

3.4.4.3 Service procedures

The procedures are defined in terms of

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

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

3.4.5 Overview

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

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

3.4.6 General conventions

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

3.4.7 Conventions for class definitions

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

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

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

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

(3) The "CLASS ID:" entry is a number that identifies the class being specified. This number is unique within the FAL ASE that will provide the services for this class. When qualified by the identity of its FAL ASE, it unambiguously identifies the class within the scope of the FAL. The value "NULL" indicates that the class cannot be instantiated. Class IDs between 1 and 255 are reserved by this standard to identify standardized classes. They have been assigned to maintain compatibility with existing national standards. CLASS IDs between 256 and 2 048 are allocated for identifying user defined classes.

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

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

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

a) Each of the attribute entries contains a line number in column 1, a mandatory (m) / optional (o) / conditional (c) / selector (s) indicator in column 2, an attribute type label

in column 3, a name or a conditional expression in column 4, and optionally a list of enumerated values in column 5. In the column following the list of values, the default value for the attribute may be specified.

b) Objects are normally identified by a numeric identifier or by an object name, or by both. In the class templates, these key attributes are defined under the key attribute.

c) The line number defines the sequence and the level of nesting of the line. Each nesting level is identified by period. Nesting is used to specify

i) fields of a structured attribute (4.1, 4.2, 4.3),

ii) attributes conditional on a constraint statement (5). Attributes may be mandatory (5.1) or optional (5.2) if the constraint is true. Not all optional attributes require constraint statements as does the attribute defined in (5.2)

iii) the selection fields of a choice type attribute (6.1 and 6.2).

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

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

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

c) The label "MgtService" designates an management service (2).

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

3.4.8 Conventions for service definitions

3.4.8.1 General

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

3.4.8.2 Service parameters

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

The service specifications of this standard uses a tabular format to describe the component parameters of the ASE service primitives. The parameters which apply to each group of service primitives are set out in tables. Each table consists of up to five columns for the

1) Parameter name,
2) request primitive,
3) indication primitive,
4) response primitive, and
5) confirm primitive.

One parameter (or component of it) is listed in each row of each table. Under the appropriate service primitive columns, a code is used to specify the type of usage of the parameter on the primitive specified in the column:

M parameter is mandatory for the primitive

U parameter is a User option, and may or may not be provided depending on dynamic usage of the service user. When not provided, a default value for the parameter is assumed.

C parameter is conditional upon other parameters or upon the environment of the service user.

— (blank) parameter is never present.

S parameter is a selected item.

Some entries are further qualified by items in brackets. These may be

a) a parameter-specific constraint:
"(=)" indicates that the parameter is semantically equivalent to the parameter in the service primitive to its immediate left in the table.

b) an indication that some note applies to the entry:
"(n)" indicates that the following note "n" contains additional information pertaining to the parameter and its use.

3.4.8.3 Service procedures

The procedures are defined in terms of

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

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

4 Concepts

IEC/TR 61158-1, Clause 9 describes the concepts of the application layer service descriptions and the templates used in this standard.

5 Data type ASE

Fieldbus data types specify the machine independent syntax for application data conveyed by FAL services. The fieldbus application layer supports the definition and transfer of both basic and constructed data types. Clause 5 of IEC 61158-5-10 provides a collection of all data types required by FAL services of type 3 and type 10 as well as data types allowed to use by application profiles. Encoding rules for the FAL data types used by type 3 are provided in IEC 61158-6-3.

6 Communication model specification

6.1 DP concepts

6.1.1 User requirements of the fieldbus DP system

The typical automation system consists of one or several programmable controllers connected by I/O-systems to the machine/process. The I/O-system is mostly implemented as a set of single point connections with a well defined electrical interface such as 24-V digital signals or 4 to 20-mA analogue signals. The overall structure of the programmable controller is defined in IEC 61131-1 and widely used.

To switch from I/O-systems plugged in a programmable controller to decentralized I/O-systems a serial link is necessary not only for the exchange of process data, but also to transmit diagnosis, configuration data and parameters.

Furthermore many applications in factory automation require very short reaction times.

Therefore the user interface in the programmable controller for the exchange of process data should be designed in a way to allow direct access to these variables without specific function calls.

In addition a reliable data transfer is required for industrial applications.

The integration of simple field devices requires a simple protocol architecture to allow low cost implementations.

To support the integration of smart field devices acyclic communication services and alarms are needed.

In order to achieve an automation system consisting of devices from different manufacturers strict interoperability of all components is necessary.

In accordance with the IEC 61131-1 structure the decentralized structure should be invisible to an existing control program.

For efficient communication between DP-slaves, as required in motion control applications, the fieldbus provides a Publisher/Subscriber communication mechanism.

The fieldbus provides a mechanism for the synchronization of the application of the DP-master (Class 1) and the DP-slaves with the communication cycle for highly synchronized real time applications.

For time stamping and other actions with precise timing requirements a mechanism to synchronies clocks is offered.

Field devices requires a mechanism to upload (Pull) and/or to download (Push) unstructured LR Data and a mechanism to invoke predefined functions within the DP-slave.

Fieldbus DP provides mechanism to build redundant DP-slaves, which may operate in control systems with several master implementations.

6.1.2 Features of fieldbus DP

Combining the structure defined in IEC 61131-1 with fieldbus DP the following advantages are achieved.

— Changes in the plant cause only small changes in wiring.

— Critical signals can be transferred over short distances to the remote field device (sensor/actuator) and to/from the programmable controller over long distances via fieldbus DP with high accuracy.

— Parameterization and diagnosis can be done without additional wiring.

— Commissioning of subparts without additional installations.

Table 1 gives an overview of the requirements and features of a fieldbus DP system.

Table 1 – Requirements and features of fieldbus DP

Requirement	Feature of fieldbus DP
Short reaction time	Exchange of more than 1 000 Inputs and Outputs with 32 field devices in less than 10 ms
Automation system consists of one or several programmable controller	Mono-master or Multi-master Operation
Simple field devices	Simple protocol, low cost communication interface
Reliable data transfer	Hamming Distance 4 guaranteed; Error recovery mechanism; Use of proven transmission technology
Excellent diagnosis	Various diagnosis in Masters as well as in Slaves
User interface with direct access to the input and output data	Buffer interface for the input and output data
Smart field devices	Acyclic communication which provides a flexible and enhanced addressing scheme of data within field devices. The possibility to transfer an alarm from the field device to the programmable controller with an explicit acknowledgement
Interoperability	Precise and complete definitions including the definition of the system behavior
Efficient communication between DP-slaves for motion control applications	The Publisher/Subscriber communication mechanism allows direct communication between DP-slaves without interactions from the User of the DP-master (Class 1)
Synchronization of the application of the DP-master (Class 1) and the DP-slaves	The Isochronous Mode provides a mechanism for the synchronization of the application of the DP-master (Class 1) and the DP-slaves with the communication cycle. The cycle jitter is less than 1 µs for data rates above 1,5 Mbit/s
Mechanism to load unstructured data blocks	To load unstructured data blocks fieldbus DP provides the Load Region mechanism
Invocation of predefined functions in the DP-slave	With the Function Invocation model predefined functions within the DP-slave may be invoked
Synchronization of clocks	With the Time Synchronization an uniform Network Time (UTC) with an accuracy of less than 1 ms can be achieved

6.1.3 DP communication model

6.1.3.1 Associations

The fieldbus communication model supports the communication of field devices with a single controlling device (e.g. programmable controller or distributed control system) via associations (see Figure 1). Furthermore, the communication between several controlling devices and their assigned field devices is also supported (see Figure 2). In addition to these basic communication models, fieldbus supports also the cyclic communication between field devices (see Figure 3).

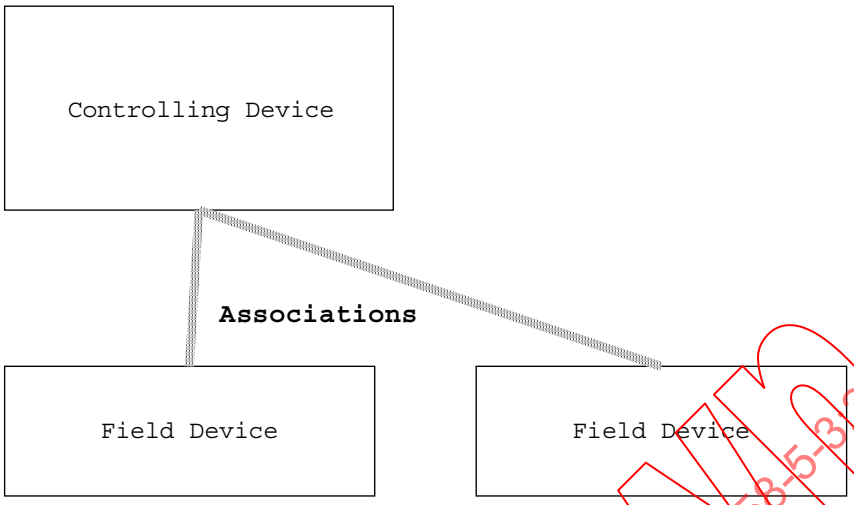

Figure 1 – Example of DP communication with a single controlling device

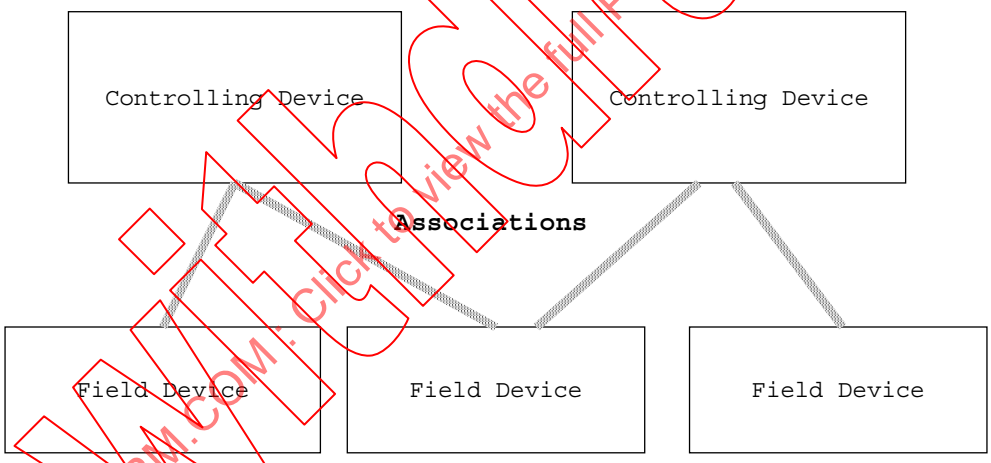

Figure 2 – Example of DP communication with several controlling devices

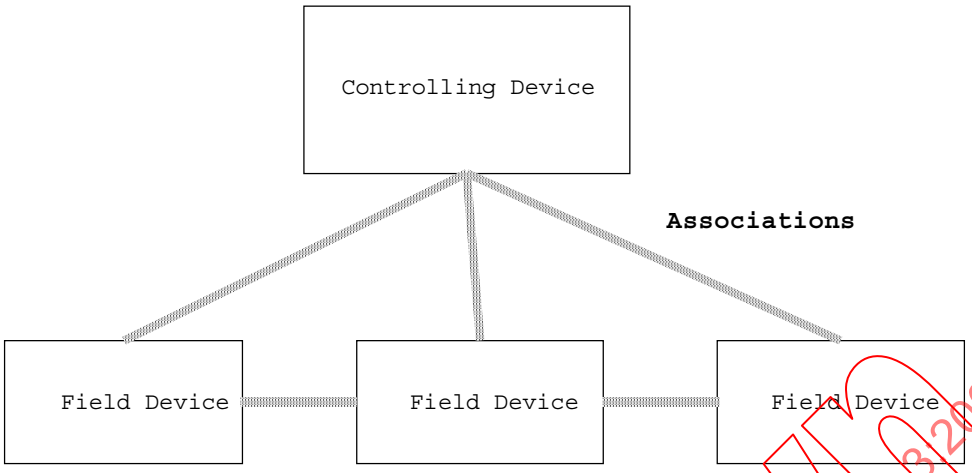

Figure 3 – Example of DP communication between field devices

6.1.3.2 Device types

6.1.3.2.1 DP-master (class 1)

The DP-master (Class 1) is a controlling device which is associated with one or more DP-slaves (field devices). The DP-master (Class 1) performs one or more of the following basic functionalities:

— cyclic exchange of I/O data with related DP-slaves
— diagnosis with the DP-slaves
— configuration of DP-slaves
— treatment of configuration and diagnosis requests of a DP-master (Class 2).

The extended functionalities of a DP-master (Class 1) are:

- acyclic access to process data of the DP-slaves
- treatment of alarms from DP-slaves
- support of the Isochronous Mode for synchronization purposes with DP-slaves
- support of the DXB mechanism for cyclic data exchange between DP-slaves
- support of the up- and/or download of LR Data with DP-slaves
- invocation of predefined functions within DP-slaves

support for clock synchronisation of the DP-slave's and other DP-master's clocks.

6.1.3.2.2 DP-master (class 2)

The DP-master (Class 2) is a controlling device which manages provision of configuration data (parameter sets) and collection of Diagnosis data for/from a DP-master (Class 1). Additionally the DP-master (Class 2) can perform all basic and some extended communication capabilities of a DP-master (Class 1) to a DP-slave.

6.1.3.2.3 DP-slave

6.1.3.2.3.1 General

The DP-slave is a field device and performs the following activities depending on the functionality.

The basic functionalities are:

- cyclic exchange of I/O data with the assigned DP-master
- diagnosis with the assigned DP-master

treatment of configuration requests of a DP-master

The extended functionalities of a DP-slave are:

— provision of acyclic access to process data for DP-masters

— provision of alarms to the assigned DP-master

— support of the Isochronous Mode for synchronization purposes with the DP-master (Class 1)

— cyclic data exchange between DP-slaves using the Publisher/Subscriber communication

— support of the up- and/or download of LR Data

— support of predefined functions which may be invoked by the DP-master (Class 1) or the DP-master (Class 2)

— provision to synchronies the local clock with a DP-master's clock

— support of DP-slave Redundancy

DP-slaves are composed of one or more modules. These modules usually reflect the hardware components of the field device.

6.1.3.2.3.2 Module

The DP Application Layer uses modules to reflect the structuring of components (e.g. hardware units, logical units) inside a DP-slave. A Module may have one or more channels which represent the real structure of the Input and/or Output Data. These channels may be a further subdivision of the Input and/or Output Data object.

Each module will be addressed by a slot number (1 to 254). Numbering is without gaps, ascending, beginning with 1. If a slot is not occupied with a module, an empty slot will be registered under the corresponding slot number in the configuration. The slot number "zero" (Slot = 0) refers to the DP-slave itself. Modules contain data which is index addressable. The concrete use of slot number and index to address data within the device is manufacturer specific.

Furthermore to each module a Configuration Identifier has to be assigned. Numbering is without gaps, ascending, beginning with 0. If a slot is not occupied with a module, a Cfg Identifier with the Input and Output Data length equal zero has to be assigned in the configuration (empty slot).

A module provides one or several of the following data elements:

- Diagnosis
  - I/O Data
  - Alarm
  - Process Data
  - LR Data
  - Context
  - Function Invocation Data

In modular DP-slaves modules determine the actual configuration of the field device (see Figure 4). The number of modules is fixed according to the device specification (maximum number of modules is 244).

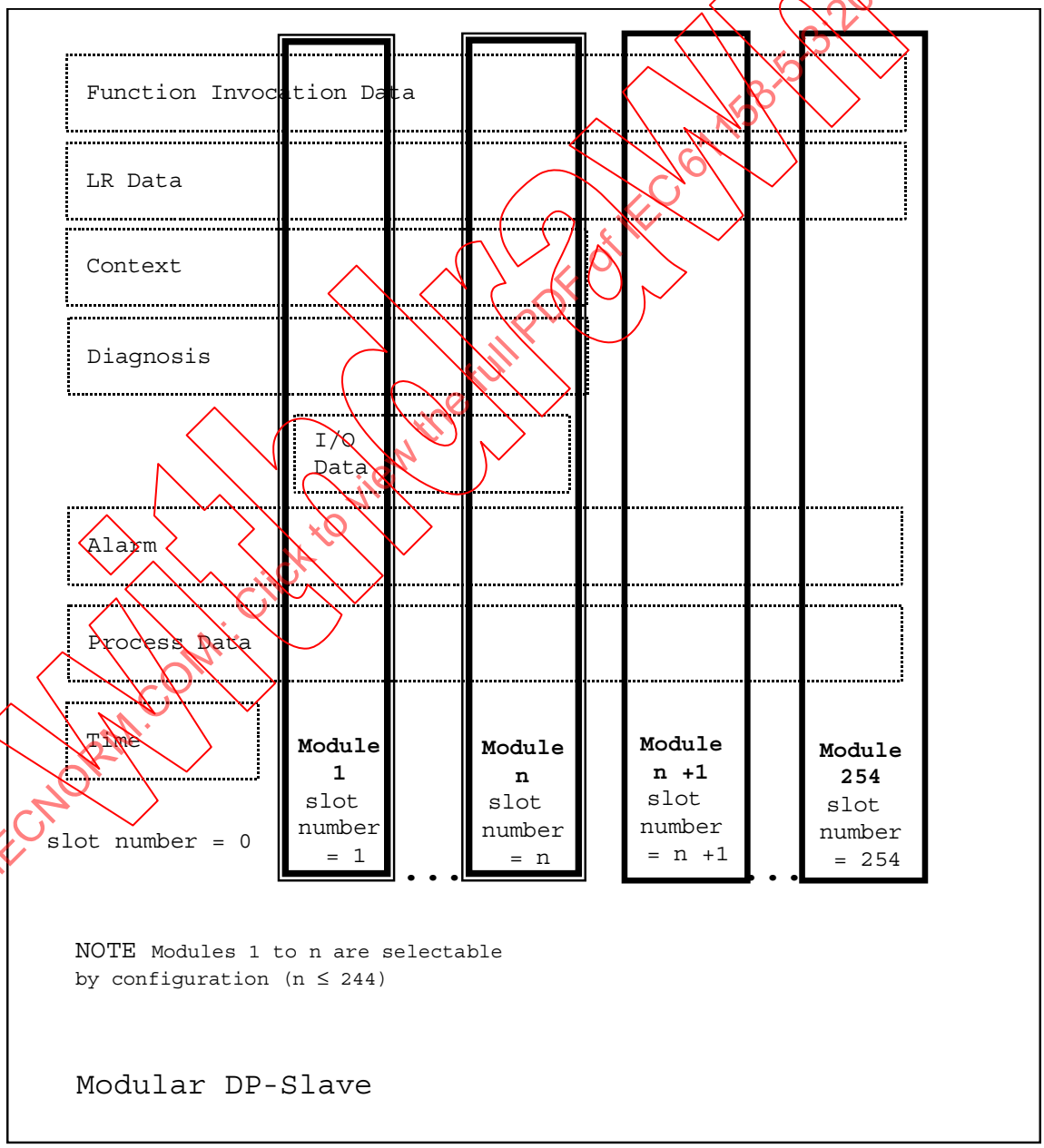

Figure 4 – DP-slave model (modular DP-slave)

In compact DP-slaves with respect to the device specification the number of modules is fixed and therefore also the configuration (for example only one module, see Figure 5).

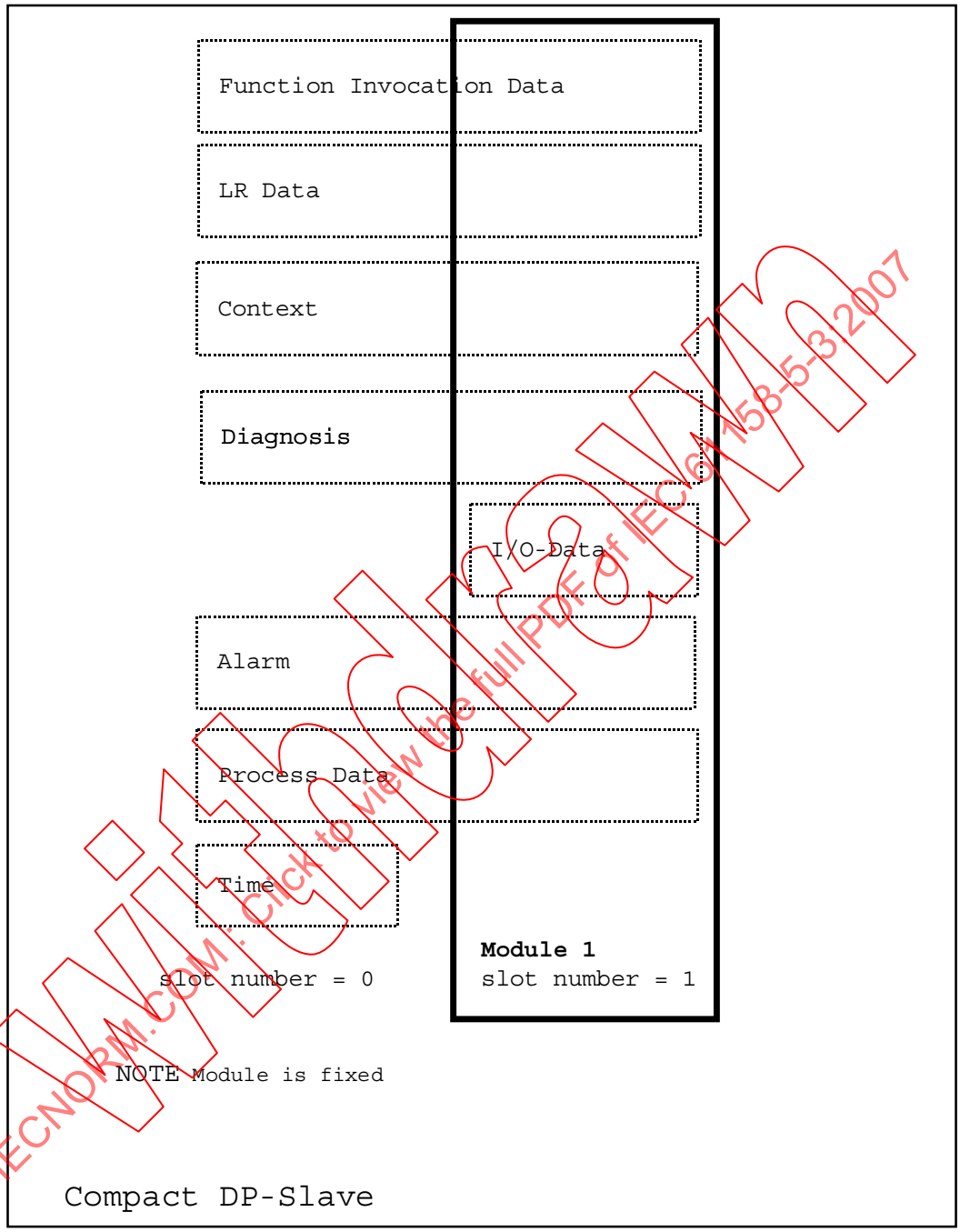

Figure 5 – DP-slave model (compact DP-slave)

6.1.3.3 Device attributes

Device attributes indicate that besides the basic functionality, extended functionality will be supported additionally.

These additional functionalities can imply certain rules for already defined data structures. A device attribute indicates that the device supports additional communication relationships or different communication behavior.

6.1.3.4 Application process

6.1.3.4.1 Overview

In the fieldbus DP environment, an application may be partitioned and distributed to a number of devices on the network. Each of these partitions is referred to as an application process (AP), (see Figure 6). A device may have several APs. In this case each individual AP is uniquely identified by an AP Identifier (API).

Every DP-slave shall have a Default AP with API = 0. The Default AP may be related to I/O data, to diagnosis and alarms, to configuration, to process data and to LR Data. All other DP-slave APs shall be related to process data only.

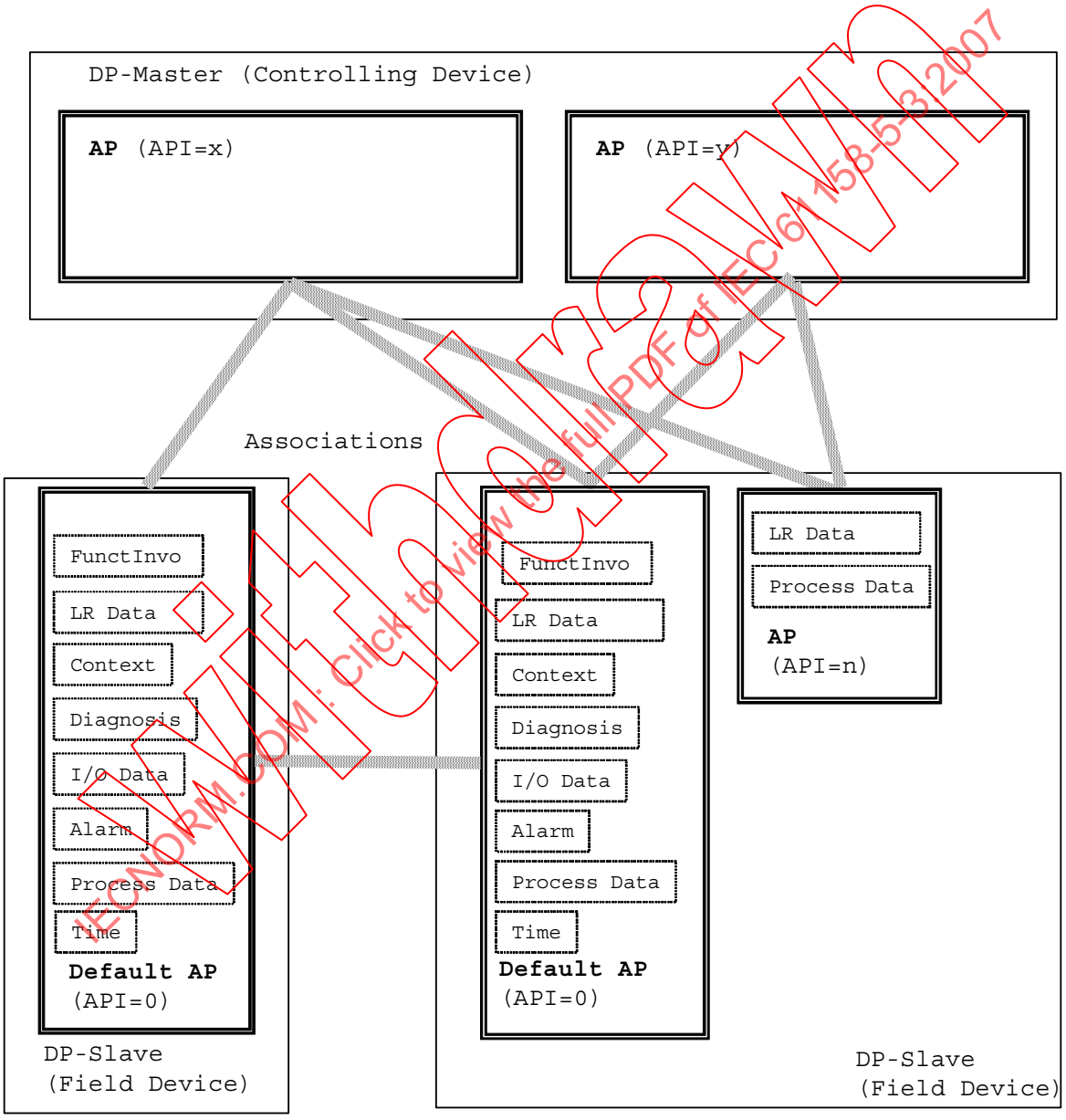

Figure 6 – Overview of application processes

An AP may be distributed to several modules. Each module within an application process is identified uniquely by a slot number (Slot = 1 to 254). The slot number "zero" (Slot = 0) refers to the AP itself within the field device.

The Context shall be related to the Default AP itself and to all its modules containing I/O Data.

The Diagnosis information may only be related to the Default AP itself and to all its modules containing I/O Data.

The I/O Data shall be related to modules of the Default AP and not to the AP itself. The number of modules with I/O Data is limited and depends on the configuration and I/O Data structure of the modules (see 6.1.3.2.3.2).

The Alarm data may be related to the Default AP itself and to all its modules.

The Process Data may be related to all APs and to all their modules.

The LR Data may be related to all APs and to all their modules.

The Time shall be related to the Default AP itself and not to its modules.

Figure 7 shows the relationship between the APs, the data elements and the modules of a DP-slave.

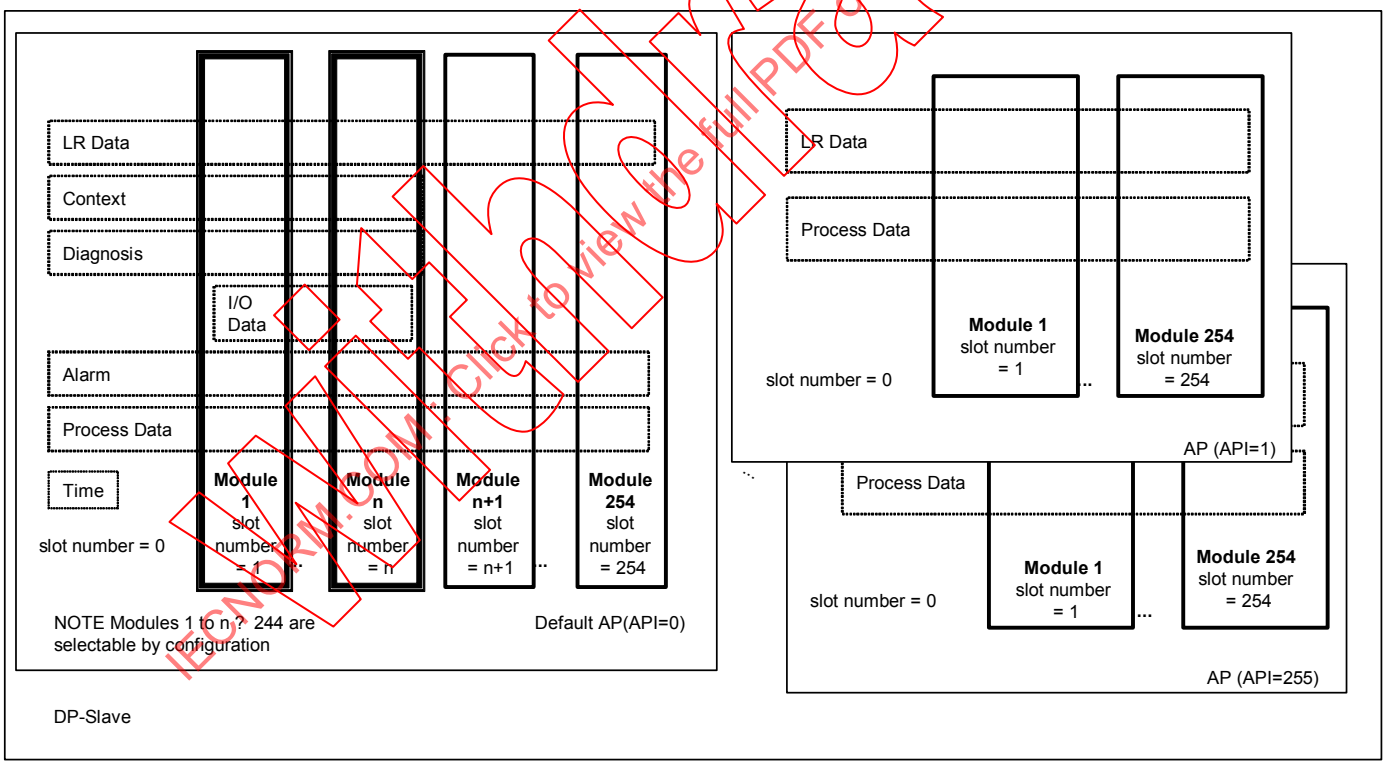

Figure 7 – DP-slave model (modular DP-slave)

The DP-master (Class 1) shall have only its Default AP. The DP-master (Class 1) shall communicate with the Default AP of a DP-slave only.

The DP-master (Class 2) may have several APs. Communicating with a DP-slave the DP-master (Class 2) may be able to address all APs of a DP-slave.

6.1.3.4.2 Application service element

An application service element (ASE), as defined in ISO/IEC 9545, is a set of application functions that provide a capability for the interworking of application processes for a specific purpose. ASEs provide a set of services for conveying requests and responses to and from application processes and their objects.

The DP Application Layer offers the following ASEs (see Figure 8):

I/O Data ASE
The I/O Data ASE provides a set of services to convey I/O data cyclically. These data always belong to those modules that have been configured in terms of the Context ASE. A multicast service provides the synchronization of the I/O data and reports the operational state of the assigned DP-master. Optionally the I/O Data ASE offers the possibility to share the input Data of one DP-slave with other DP-slaves.

Process Data ASE
The Process Data ASE provides a set of services to convey acyclically data. The application of the DP-master requests each transmission individually. The Process Data ASE can be related to all APs of a DP-slave.

Diagnosis ASE
The Diagnosis ASE provides services for the DP-master to fetch Diagnosis information from a DP-slave which is related to the Default AP or to its modules that are part of the configuration.

Alarm ASE
The Alarm ASE provides a set of services to convey alarms or status messages issued by the DP-slave. The assigned DP-master acknowledges the Alarms. Status messages will not have to be acknowledged by the DP-master.

Management ASE
The Management ASE provides a set of services to manage a DP-master (Class 1) by a DP-master (Class 2). The DP-master (Class 1) receives the bus parameter set or the slave parameter set. Afterwards the received sets will be activated by the DP-master (Class 2). The DP-master (Class 1) will provide the system diagnosis information if requested.

Context ASE
The Context ASE provides a set of services to

- parameterize and configure the Default AP itself and its modules containing I/O data
- transfer switchover commands for redundant structures and commands to reset MS1 or change operation mode to the DP-slave
- set the device address of a DP-slave
- to provide the real configuration of the DP-slave to be read by a DP-master (Class 2)

and to establish or release an association between an individual AP of a DP-master (Class 2) and an individual AP of the DP-slave for acyclic communication. This association exclusively belongs to the Process Data ASE.

Load Region ASE
The Load Region ASE provides a set of services to convey LR Data acyclically. The application of the DP-master requests each transmission individually. The Load Region ASE can be related to all APs of a DP-slave.

Function Invocation ASE
The Function Invocation ASE provides a set of services to invoke stateless and/or state-oriented Function Invocation objects. The application of the DP-master has to invoke each function individually. The Function Invocation ASE can be related to all APs of a DP-slave.

Time ASE
The Time ASE provides a set of services to synchronies the clocks of several or all devices on a fieldbus segment. The application of the DP-master (Class 1) requests each synchronization individually. The Time ASE can be related to the Default AP of a DP-slave.

Application Relationship ASE
The Application Relationship ASE (AR ASE) provides a description model for the separate AR types. This includes their transfer characteristics as well as their current communication states.

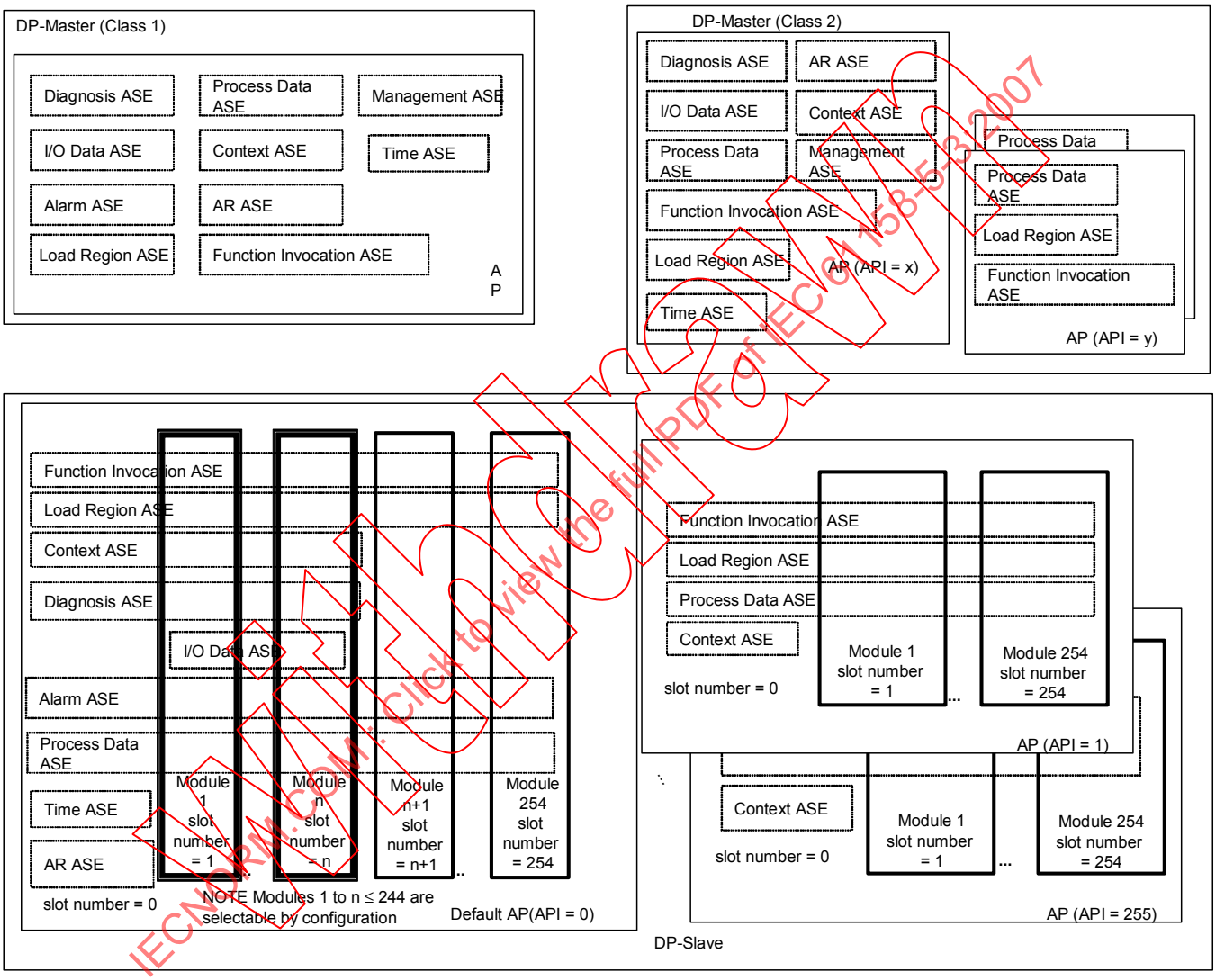

Figure 8 – Application Service Elements (ASEs)

6.1.3.4.3 Application objects

An application object (APO) is a network representation of a specific aspect of an AP. Each APO represents a set of information and processing capabilities of an AP that are accessible through services of the DP Application Layer. APOs are used to represent these capabilities to other APs in a fieldbus DP system.

In order to permit an AP to communicate with an AP of an another device, APOs have to be available. Application objects (virtual objects) represent existing process objects (real

objects), that an application process has made visible and accessible to the communication (see Figure 9).

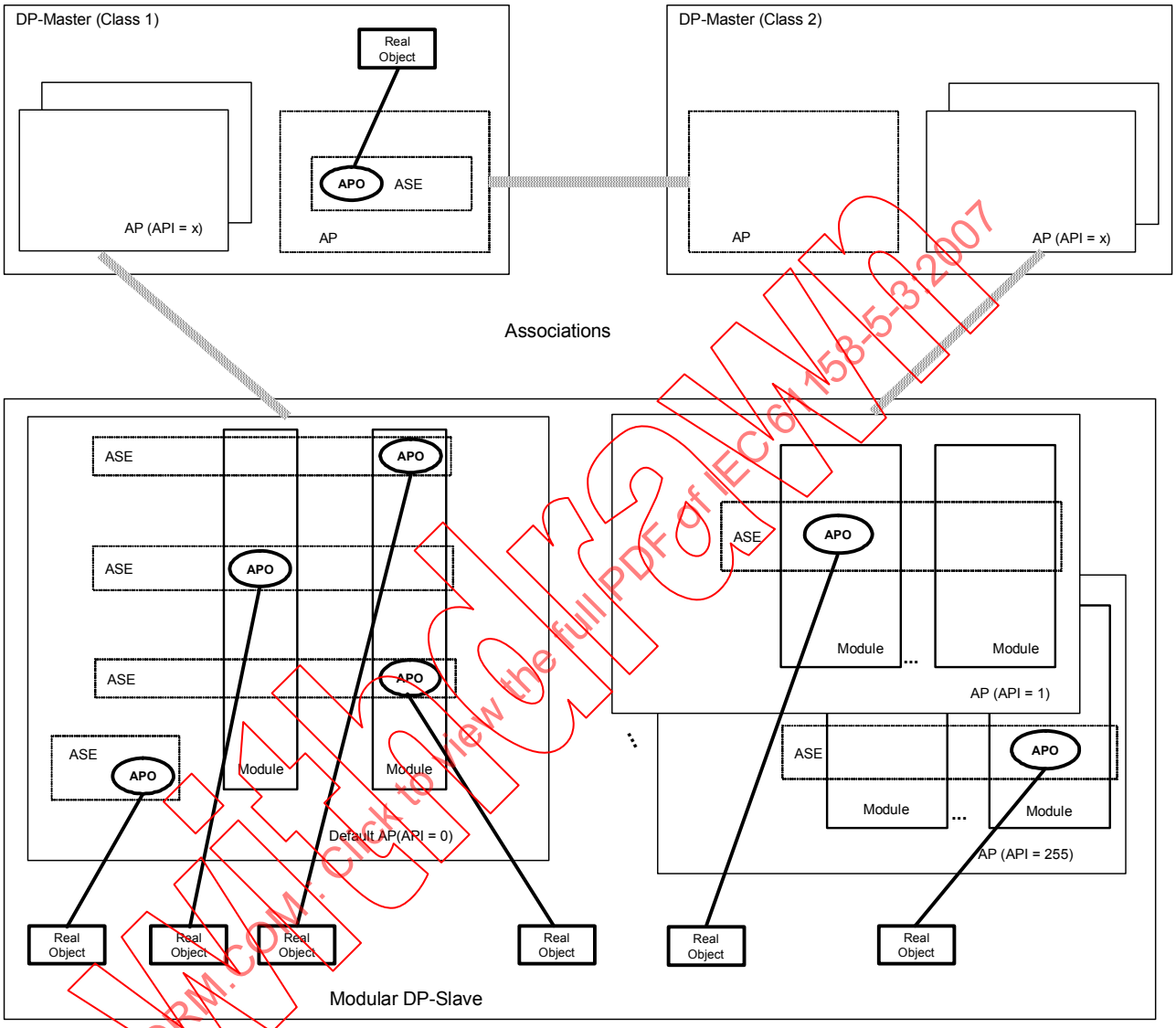

Figure 9 – Application Process with application Objects (APOs)

In Figure 10, a remote AP acting as Client may access the real object by sending requests through the APO that represents the real object. Local aspects of the AP convert between the network view (the APO) of the real object and the internal AP view of the real object.

Within the AP an APO is identified by slot and index. The address space which is defined by slot and index can be used by several APs.

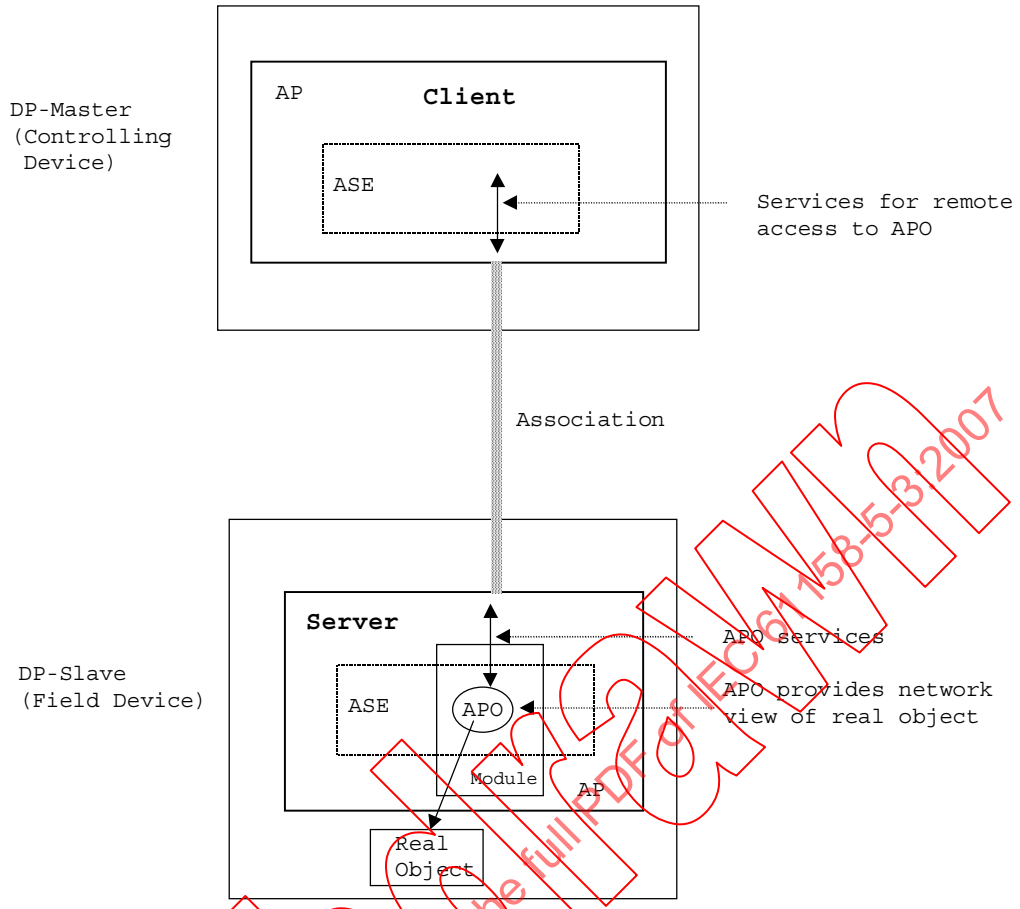

Figure 10 – Access to a remote APO

In Figure 11, a Client/Server and a Publisher/Subscriber association is shown. The Client may access the real object by sending requests through the APO that represents the real object. The Subscriber may subscribe to the whole or to a part of the remote APO. The Publisher/Subscriber association is always related to the Default AP and the I/O Data ASE. Local aspects of the AP convert between the network view (the APO) of the real object and the internal AP view of the real object.

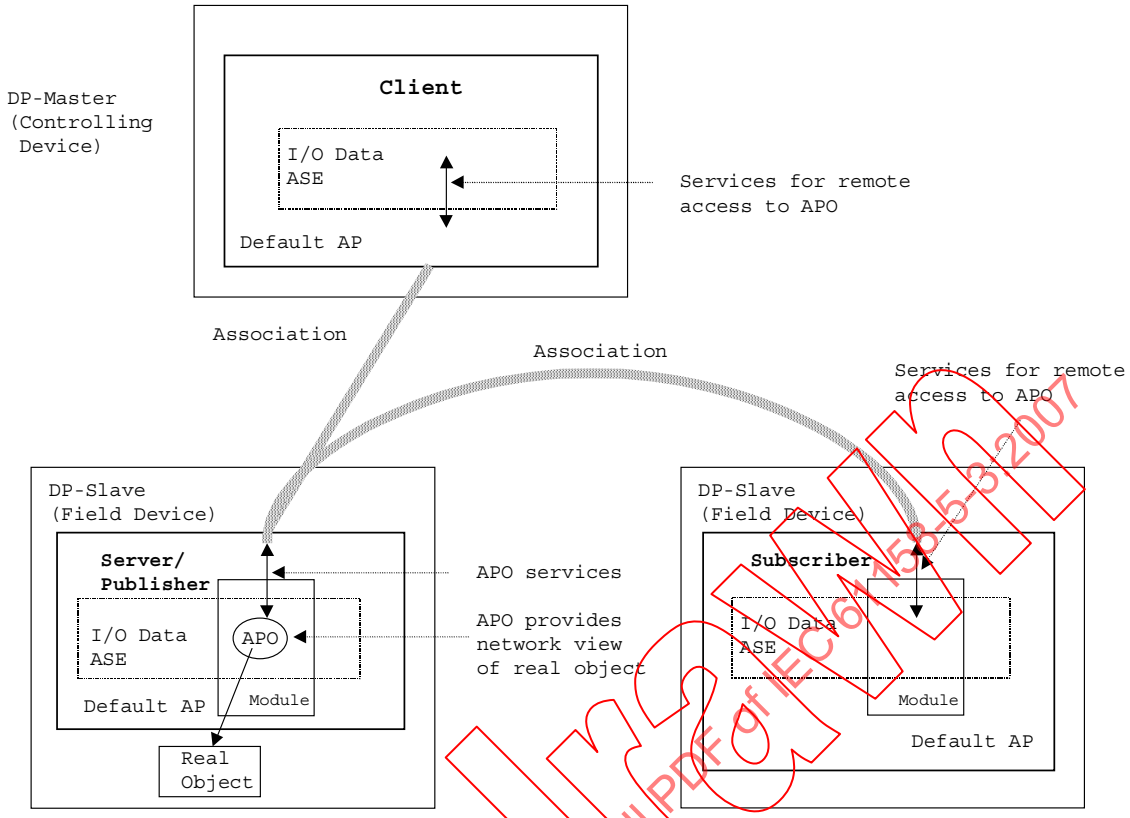

Figure 11 – Access to a remote APO for publisher/subscriber association

6.1.3.5 Application relationship

6.1.3.5.1 General

An Application Relationship (AR) is a co-operative relationship between two or more APs for the purpose of exchange of information and co-ordination of their joint operation (see Figure 12). This relationship is activated by the exchange of Application Protocol Data Units (APDU). Fieldbus DP uses different types of ARs which distinguish in their conveyance characteristic.

6.1.3.5.2 Application relationship endpoint

An AP has access to the communication using application relationship endpoints (AREP, see ISO 7498-3). One or more AREPs are fixed and uniquely assigned to an AP. These endpoints are addressed by the AP through an identifier (AREP ID). These identifiers are device specific and not defined by the communication itself. ARs are defined as a set of co-operating AREPs. Between two APs one or more ARs may exist, each one having unique AREPs. Figure 12 shows an example of one AR with two AREPs.

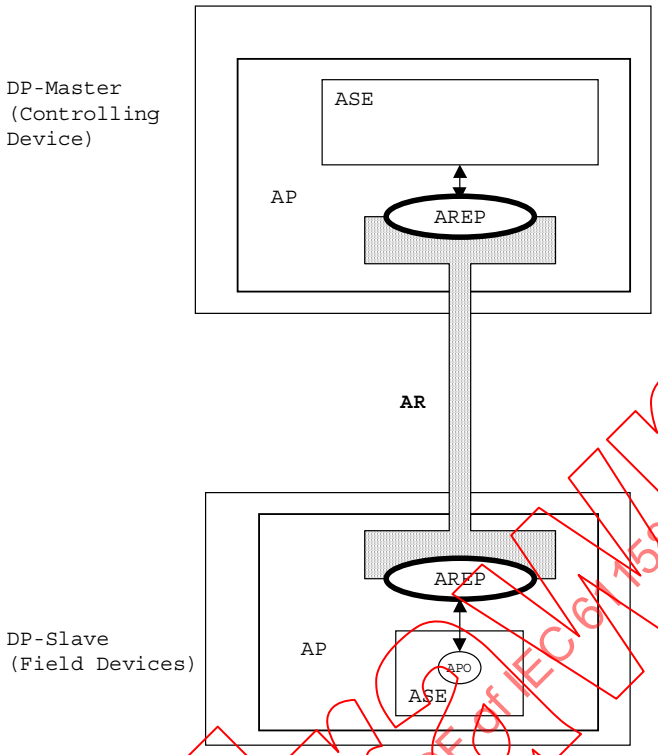

Figure 12 – Example of one AR with two AREPs

6.1.3.5.3 Overview of application relationships

The DP Application Layer offers the following AR types:

MS0: application relationship between the application process of one DP-master (Class 1) and all related DP-slaves and optional between the AP of one or several DP-master (Class 2) and all related DP-slaves and optional between the AP of one or several DP-slaves with all related DP-slaves for the following purposes:

- cyclic exchange of the I/O data with the DP-master (Class 1)
- cyclic exchange of the Input Data between DP-slaves (DXB)
- acyclic data transfer for parameterization, configuration and diagnosis (DP-master (Class 1))
- acyclic transfer of commands to a set of field devices (DP-master (Class 1))
- cyclic transfer of synchronization messages to a set of field devices (DP-master (Class 1))
- acyclic read of the I/O data (DP-master (Class 2))
- acyclic read of configuration information (DP-master (Class 2))
- acyclic read of diagnosis information (DP-master (Class 2))

acyclic write of remanent parameter (DP-master (Class 2))

MS1: connection-oriented relationship between the application processes of one DP-master (Class 1) and of one related DP-slave for the following purposes:

- acyclic read and write of variables
- acyclic transfer of alarms
- up- and/or download of LR Data

invocation of stateless and/or state-oriented functions

MS2: connection-oriented relationship between the application processes of one DP-master (Class 2) and of one related DP-slave for the purpose:

- acyclic read and write of variables
- up- and/or download of LR Data

invocation of stateless and/or state-oriented functions

MS3: connectionless relationship between the application processes of one DP-master and of a set of related DP-slaves for the purpose:

– synchronisation of time

MM1: connectionless relationship between the application processes of one configuration device (Master Class 2) and of one related controlling device (Master Class 1) for the following purposes:

- upload and download for configuration information
- upload of diagnosis information

activation of the previously transferred configuration

MM2: connectionless relationship between the application processes of one configuration device (Master Class 2) and of a set of related controlling devices (Master Class 1) for the purpose:

activation of the previously transferred configurations

6.1.4 Dynamical behavior of fieldbus DP

Before commissioning a DP system, all stations shall have been assigned an unique address. In the case of DP-slaves, this address may be set up using the MS0 AR (see 6.2.5.3.9).

All DP-slaves that have not yet been assigned an individual address, shall start with the default address 126. Only one device with this address shall be allowed on the network. A DP-master address shall not be set to the default address. A DP-master (Class 2) may access a DP-slave via the default address and may assign a specific address to that device.

For security reasons, the DP-master (Class 1) shall not exchange I/O data with a DP-slave whose address is 126.

To exchange I/O data a DP-master (Class 1) shall have a valid Master Parameter Set which may be loaded and activated by the DP-master (Class 2) via the MM1 AR and MM2 AR. The Master Parameter Set consists of several sets of configuration data and parameters, each dedicated to a DP-slave. This set of data shall contain all necessary information (for the DP system) to describe a DP-slave. Additionally, the Master Parameter Set shall include the bus parameters as well as the address assignment table which assigns each individual remote I/O data a unique address in the I/O space of the DP-master's User.

If there exists a valid Master Parameter Set in a DP-master (Class 1), this device starts to check whether the DP-slaves dedicated to that DP-master are present or not (this will be done

with a read of the diagnosis of a DP-slave). After an appropriate answer the DP-master (Class 1) will set the parameters and check the configuration of that DP-slave. A DP-slave will accept a check configuration request only from the Master who sets the parameters.

After submitting these two functions correctly, a DP-master (Class 1) checks the status of the DP-slave by reading the Diagnosis information. If the configuration check was successful and parameterization was correct, the DP-master (Class 1) will enter the user data exchange mode. A DP-slave accepts data exchange requests only from a DP-master (Class 1) which has previously submitted parameters and configuration. If a diagnosis event occurred in the application process of a DP-slave, this event is signaled by sending a response with high priority. The DP-master (Class 1) will read the diagnosis and inform its local User.

The DP-master (Class 1) sends information about his own operation mode (CLEAR, OPERATE) to his dedicated DP-slaves for synchronization. This will be done at every change of the operation mode of the DP-master (Class 1) and additionally at regular intervals (Data Control Time/2).

Only in the data exchange mode the extended functionalities like the acyclic read and write of variables, the acyclic transfer of alarms, the up- and/or download of LR Data, the invocation of stateless and/or state-oriented functions, the DXB functionality and the Isochronous Mode are possible if supported by both the DP-master (Class 1) and the DP-slave. Errors on the acyclic communication (MS1 AR) also influence the cyclic data transfer (MS0 AR) and vice versa. This means that both communication relationships will be aborted. Each alarm has to be acknowledged by the assigned DP-master (Class 1).

Every DP-master (Class 2) can read the diagnosis information, the configuration, the input and output data (via the MS0 AR) of every DP-slave although the DP-slave is under control of a DP-master (Class 1). A DP-master (Class 2) can take a DP-slave under its control. In this case the DP-slave will stop the data exchange mode with the DP-master (Class 1). In this case the DP-master (Class 1) starts to read the diagnosis cyclically. If the DP-master (Class 2) has finished its communication with this DP-slave, the DP-slave will set the field "Master Add(ress)" to invalid. This event triggers the DP-master (Class 1) to gain control over this DP-slave again. As described above it will send parameters and configuration first.

If supported by both the DP-master (Class 2) and the DP-slave the extended functionality like the acyclic read and write of variables, the up- and/or download of LR Data and the invocation of stateless and/or state-oriented functions is possible (via the MS2 AR). The MS2 AR is independent of the MS0 AR and MS1 AR.

Fieldbus DP defines a redundancy model with the following features:

- One slave redundancy model for all redundancy structures (even for non-redundant systems)
- Scalability: Independent master, slave, and line redundancy
- Easy engineering: No additional user efforts, no complex tools necessary
- Complete monitoring of all components
- No influence on bus load and timing
- High reliability
- Short switch over time

No loss of data during fault tolerance

A redundant DP-slave may consist of two communication interfaces with special redundancy extensions and an independent redundancy communication channel between these two communication interfaces. One of the slave interface modules acts as the primary and one as the backup slave.

The redundancy model of fieldbus DP is based on a switchover command, commands to stop, start and reset the MS1 AR, provided by the MS0 AR via the Check User Prm service to the application of the DP-slave and a Diagnosis object (Red Status) which provides information about the state of the redundant DP-slave if it is primary or backup and if a previous provided command (switchover, start, stop reset of the MS1 AR) has been processed.

During the processing of a switchover command or a stop, start or reset command the MS0 AR remains in the data exchange mode. Only the MS1 AR will be influenced, therefore a seamless takeover of a DP-slave may be accomplished.

The communication with a redundant DP-slave will be done in the following way:

MS0 AR:

The primary slave uses the MS0 AR (cyclic communication) to send and receive the I/O data. In addition the primary slave sends all diagnosis information about itself and the backup slave.

The backup slave may also send and receive values for the I/O data via the MS0 AR, but these values are not relevant and shall be ignored by the backup slave. The backup slave may transmit its own diagnosis information, but, it shall also be part of the primary diagnosis.

MS1 AR:

Only the primary slave will receive and transmit MS1 AR services. When the backup slave receives a MS1 AR service, it shall respond negatively.

MS2 AR:

Both the primary and the backup slave may receive acyclic MS2 AR services. Both slave interface modules shall process these services to allow an individual communication with the specific device. If values will be sent with MS2 AR services, these data may be sent to one slave interface module only and shall be exchanged with the other using the redundancy communication channel between the two communication interfaces. Only data, which will be held in both slave interface modules, shall be exchanged to keep them consistent.

The features for redundancy could be used for systems with backup masters as well and provide means for continuous cyclic operation.

6.1.5 Fieldbus DP status values

Table 2 shows the Status Values of the service primitives.

Table 2 – Status values of the service primitives

Short Name	Meaning
ABT_FE	Invalid request APDU received
ABT_IA	Invalid additional address information
ABT_IV	Invalid service from the User
ABT_OC	Waiting for DL-DATA-REPLY.cnf
ABT_RE	Invalid response APDU received
ABT_SE	Sequence error; service not allowed in this state
ABT_STO	Send Timeout requested was too small
ABT_TO	Timeout of the connection
AD	Access denied
DH	Positive acknowledgement for sent data, reply data with high priority available
DI	Data Incomplete
DS	Local-DL/PHY Entity is not in logical token-ring or disconnected from line
EA	Area too large (Up-/Download)
FE	Format-Error in a Request-APDU
GE	Group Error, invalid group at service request (Global Control)
IP	Invalid Parameter
IV	Invalid parameters in request
LE	Data-block-length too large (Up-/Download)
LR	Resources local not available or not sufficient
LS	Service at local DLSAP or local DLSAP not activated.
NA	Negative ack, no reaction from remote station
NC	Master Parameter Set not compatible
NE	Area non-existent
NI	Service not implemented
NO	Service in this state not possible
NR	No Response data
RDH	Response data high and no resource for sent data
RDL	Response data low and no resource for sent data
RE	Format-Error in a Response-APDU
RR	Resources of the remote-DL Entity not sufficient or not available
RS	Service or remote-address at remote-DLSAP or remote-DLSAP not activated - remote-station is no DP-Station - remote-station is not yet ready for these Service - remote-station is associated with an other Requester - optional service not available
SC	Sequence Conflict
SV	Sequence violation (second Time Master active)
SE	Sequence Error
TO	Function-Timeout expired
UE	Remote-DMPM / DL interface error

6.2 ASEs

6.2.1 Process data ASE

6.2.1.1 Overview

In the DP Application Layer environment, application processes contain data that remote applications are able to read and write. The Process Data ASE defines attributes of general purpose Process Data objects and provides a set of services used to read, and write their values. A Process Data ASE is specific for an AP and every AP may contain one Process Data ASE. Process Data objects within an AP are addressed by Slot Numbers and Indices. The Process Data object can be read and written partially or entirely.

The DP-masters (Class 1) and the DP-master (Class 2) are able to read or write the value of the Process Data object within the Default AP of a DP-slave. Process Data objects outside the Default AP of a DP-slave can only be accessed by the DP-master (Class 2).

The DP-master (Class 2) has to use the Context ASE (see 6.2.5.3.10) to establish an association to get access to the Process Data objects of a DP-slave. The Process Data object in combination with the Context ASE provides the possibility of access protection (see 6.2.1.3).

The DP-master (Class 1) can access the values of the Process Data object within the Default AP after the DP-slave has entered the data exchange mode. This association provides no access protection for the accessible Process Data object.

The access to Process Data objects is performed according to the Client/Server access model. The Client/Server model is characterized by a Client application sending a read or write request to a Server application that responds accordingly. The Server's activity is stimulated by Clients on the network.

The formal model of the Process Data ASE is presented by the Process Data class specification, containing a description of its attributes, services as well as invocations, followed by a detailed service specification.

6.2.1.2 Process data class specification

6.2.1.2.1 General

The Process Data ASE defines two Process Data object types:

— Simple objects

— Combined objects

Simple objects may have a one to one relation to real objects. Combined objects may have a one to two relation to real objects. Combined objects may be read and written with one service interaction. This type of objects may be used for communication which requires consistency between read and write or high protocol efficiency.

6.2.1.2.2 Simple process data class specification

6.2.1.2.2.1 Template

A Simple Process Data object is always related to one real object as shown in Figure 13.

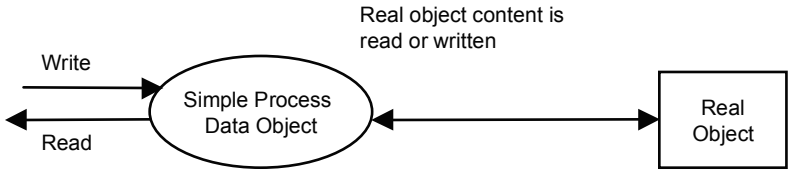

Figure 13 – Relation of a simple process data object to the real object

A Process Data object is described by the following template:

DP ASE:			**Process Data ASE**
CLASS:		**Simple Process Data**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attributes:	Identifier
2.	(m)	Attribute:	Process Data Description
3.	(m)	Attribute:	Partial Access
4.	(m)	Attribute:	Access Rights MS1
5.	(m)	Attribute:	Access Rights MS2
6.	(c)	Constraint:	Access Rights MS2 = Right to Read or Right to Read/Write
6.1.	(m)	Attribute:	List of SCL Read
6.1.1.	(m)	Attribute:	SCL
7.	(c)	Constraint:	Access Rights MS2 = Right to Write or Right to Read/Write
7.1.	(m)	Attribute:	List of SCL Write
7.1.1.	(m)	Attribute:	SCL
8.	(o)	Attribute:	Process Data Name
9.	(o)	Attribute:	Local Detail
SERVICES:			
1.	(o)	OpsService:	Read
2.	(o)	OpsService:	Write

6.2.1.2.2.2 Attributes

Identifier
This key attribute is a triplet composed of API, Slot Number, and Index to define to which Application Process Identifier, module and to which index within the specified module the Simple Process Data object belongs. This Identifier shall not be used by another Action object, Function Invocation object, Load Region object or Process Data object.

Attribute Type: Unsigned8, Unsigned8, Unsigned8

Allowed triplet values: [0 to 254, 0 to 254, 0 to 254]

Process Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Partial Access
This Boolean attribute defines whether a partial access to the Simple Process Data object is supported or not. Partial access equal TRUE means that the content of the Simple Process Data object is read or written from the first octet up to the octet defined by the length given in the service. The length given in the service shall be less or equal the value of the attribute length in case of a write access. For a read access the entire object will be read if the length given in the service is greater than the length of the object. Partial access equal FALSE means that the content of the Simple Process Data object can only be read or written entirely. That implies that the length given in the service shall be equal to or (in case of read access) larger than the length of the object.

Attribute Type: Boolean

Access Rights MS1
This attribute defines the type of access rights defined for a MS1 AR for the Simple Process Data object.

The allowed values are shown in Table 3.

Table 3 – Access Rights MS1

Value	Meaning
0	Right to Write
1	Right to Read
2	Right to Read/Write
3	No Access

Access Rights MS2
This attribute defines the type of access rights defined for a MS2 AR for the Simple Process Data object.

The allowed values are shown in Table 4.

Table 4 – Access Rights MS2

Value	Meaning
0	Right to Write
1	Right to Read
2	Right to Read/Write
3	No Access

List of SCL Read
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one octet value for read access protection to this Simple Process Data object. The rules for accepting or rejecting a read access are described in 6.2.1.3.

Attribute Type: Unsigned8

List of SCL Write
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one octet value for write access protection to this Simple Process Data object. The rules for accepting or rejecting a write access are described in 6.2.1.3.

Attribute Type: Unsigned8

Process Data Name
This optional attribute defines the name of the Process Data object.

Attribute Type: Visible String(32)

Local Detail
This optional attribute contains additional local information about the Simple Process Data object.

Attribute Type: Octet String

6.2.1.2.2.3 Invocation of the simple process data object

For the invocation of the Simple Process Data object the following rules apply:

— Simple Process Data objects shall not exceed the total length of 240 octets. The attribute Process Data description has to be set accordingly.

— At least one service shall be allowed for the access of the object.

— At least one AR shall have access to the object.

— The access rights have to be set accordingly to the allowed service(s).

6.2.1.2.3 Combined process data class specification

6.2.1.2.3.1 Template

A Combined Process Data object is always related to two real objects as shown in Figure 14.

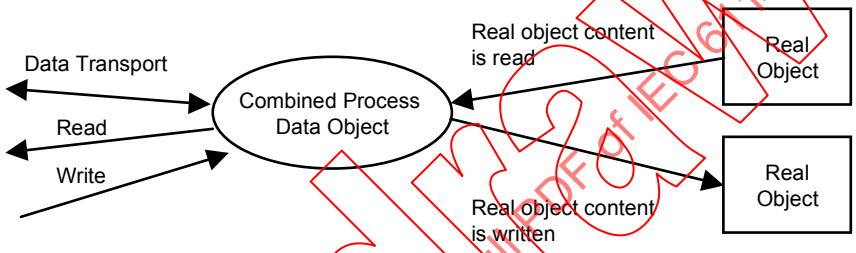

Figure 14 – Relation of a combined process data object to the real objects

A Combined Process Data object is described by the following template:

DP ASE:			**Process Data ASE**
CLASS:		**Combined Process Data**	
CLASS ID:		not used	
PARENT CLASS:		TOP	
ATTRIBUTES:			
1.	(m)	Key Attributes:	Identifier
2.	(m)	Attribute:	Read Process Data Description
3.	(m)	Attribute:	Read Partial Access
4.	(c)	Constraint:	Access Rights MS2= Right to Read or Right to Read/Write
4.1.	(m)	Attribute:	List of SCL Read
4.1.1.	(m)	Attribute:	SCL
5.	(m)	Attribute:	Write Process Data Description
6.	(m)	Attribute:	Write Partial Access
7.	(c)	Constraint:	Access Rights MS2 = Right to Write or Right to Read/Write
7.1.	(m)	Attribute:	List of SCL Write
7.1.1.	(m)	Attribute:	SCL
8.	(m)	Attribute:	Access Rights MS1
9.	(m)	Attribute:	Access Rights MS2
10.	(o)	Attribute:	Process Data Name
11.	(o)	Attribute:	Local Detail
SERVICES:			
1.	(o)	OpsService:	Data Transport
2.	(o)	OpsService:	Read
3.	(o)	OpsService:	Write

6.2.1.2.3.2 Attributes

Identifier
This key attribute is a triplet composed of API, Slot Number, and Index to define to which Application Process Identifier, module and index within the specified module the Combined Process Data object belongs. This Identifier shall not be used by another Action object, Function Invocation object, Load Region object or Process Data object.

Attribute Type: Unsigned8, Unsigned8, Unsigned8

Allowed triplet values: [0 to 254, 0 to 254, 0 to 254]

Read Process Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. of the Process Data ASE. For data description the template given in IEC 61158-5-10, 5.1 is used.

Read Partial Access
This Boolean attribute defines whether a partial read access to the Combined Process Data object is supported or not. Partial access equal TRUE means that the content of the Combined Process Data object is read from the first octet up to the octet defined by the length given in the service. The entire object will be read if the length given in the service is greater than the length of the object. Partial access equal FALSE means that the content of the Combined Process Data object can only be read entirely. That implies that the length given in the service shall be equal to or larger than the length of the object.

Attribute Type: Boolean

List of SCL Read
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one octet value for read access protection to this Combined Process Data object. The rules for accepting or rejecting a read access are described in 6.2.1.3.

Attribute Type: Unsigned8

Write Process Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Write Partial Access
This Boolean attribute defines whether a partial write access to the Combined Process Data object is supported or not. Partial access equal TRUE means that the content of the Combined Process Data object is written from the first octet up to the octet defined by the length given in the service. The length given in the service shall be less or equal the value of the attribute length. Partial access equal FALSE means that the content of the Combined Process Data object can only be written entirely. That implies that the length given in the service shall be equal to the length of the object.

Attribute Type: Boolean

List of SCL Write
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one octet value for write access protection to this Combined Process Data object. The rules for accepting or rejecting a write access are described in 6.2.1.3.

Attribute Type: Unsigned8

Access Rights MS1
This attribute defines the type of access rights defined for a MS1 AR for the Combined Process Data object.

The allowed values are shown in Table 5.

Table 5 – Access Rights MS1

Value	Meaning
0	Right to Write
1	Right to Read
2	Right to Read/Write
3	No Access

Access Rights MS2
This attribute defines the type of access rights defined for a MS2 AR for the Combined Process Data object.

The allowed values are shown in Table 6.

Table 6 – Access Rights MS2

Value	Meaning
0	Right to Write
1	Right to Read
2	Right to Read/Write
3	No Access

Process Data Name
This optional attribute defines the name of the Process Data object.

Attribute Type: Visible String(32)

Local Detail
The attribute local detail contains additional local information about the Simple Process Data object.
Attribute Type: Octet String

6.2.1.2.3.3 Invocation of the combined process data object

For the invocation of the Combined Process Data object the following rules apply:

— Combined Process Data objects shall not exceed the total length of 240 octets for both Read Process Data and Write Process Data. The attributes Read Data Description and Write Data Description have to be set accordingly.

— At least one service shall be allowed for the access of the object.

— At least one AR shall have access to the object.

— The access rights have to be set accordingly to the allowed optional service(s).

6.2.1.3 Access protection on process data objects

The Process Data ASE in combination with the Context ASE provides also access protection. This is not a protection against intentional misuse of the communication facilities of a field device but helps to protect a system for accidental erroneous use of Process Data.

The protection is accomplished by two mechanisms:

— the Initiate service will set up a security level for this CR.

— the Process Data object will have a list of SCL values for the read and write service. An access will be accepted only if the SCL of the initiate service matches (see Table 7) with an element of the SCL list for this Process Data object.

This allows the User to select a level for read and write objects. Thus a service device can use a SCL which allows access to almost all objects. A text display may have only read access to specific objects.

Table 7 – SCL matching rules

SCL in Initiate.ind	SCL in Process Data object	Condition	Result
= 0	= 0	-	access allowed
≠ 0	= 0	-	access allowed
= 0	≠ 0	-	access denied
≠ 0	≠ 0	SCL in Initiate.ind not contained in List of SCL at Process Data object	access denied
≠ 0	≠ 0	SCL in Initiate.ind contained in List of SCL at Process Data object	access allowed

6.2.1.4 Process data service specification

6.2.1.4.1 Read

This confirmed service may be used to read the value of a Process Data object. This service shallshould only be used in conjunction with the MS1 or MS2 AR. Table 8 shows the parameters of the service.

Table 8 – Read

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
Index	M	M(=)		
Length	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Length			M	M(=)
Data			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error Code 1			M	M(=)
Error Code 2			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Process Data object in a specific slot (typically a module).

Index
The parameter Index is used in the destination device for addressing the desired Process Data object.

Length
The parameter Length indicates the number of octets of a Process Data object that has to be read.

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

Data
The parameter Data contains the value of the object which has been read and consists of the number of octets indicated in the Length of the response. This parameter shall be composed of the data types defined in IEC 61158-5-10, Clause 5.

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

Error Decode
This parameter selects one of the following Error schemes: DPV1, FMS, Profile_Specific.

Error Code 1
In case of Error Decode equal DPV1 the Error Code 1 assumes one of the following values:
read error, module failure, version conflict, feature not supported, user specific, invalid index, invalid slot, type conflict, invalid area, state conflict, access denied, invalid range, invalid parameter, invalid type, read constrain conflict, resource busy, resource unavailable.

Error Code 2
The parameter Error Code 2 is user specific.

6.2.1.4.2 Write

This confirmed service may be used to write the value of a Process Data object. This service shall only be used in conjunction with the MS1 or MS2 AR. Table 9 shows the parameters of the service.

Table 9 – Write

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
Index	M	M(=)		
Length	M	M(=)		
Data	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Length			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error Code 1			M	M(=)
Error Code 2			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Process Data object in a specific slot (typically a module).

Index
The parameter Index is used in the destination device for addressing the desired Process Data object.

Length
The parameter Length indicates the number of octets of a Process Data object which has to be written.

Data
The parameter Data contains the value of the Process Data object which has to be written and consists of the number of octets indicated in the Length of the request. This parameter has to be composed of the data types defined in IEC 61158-5-10, Clause 5.

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

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

Error Decode
This parameter selects one of the following Error schemes: DPV1, FMS, Profile_Specific.

Error Code 1
In case of Error Decode equal DPV1 the Error Code 1 assumes one of the following values:
write error, module failure, version conflict, feature not supported, user specific, invalid index, write length error, invalid slot, type conflict, invalid area, state conflict, access

denied, invalid range, invalid parameter, invalid type, write constrain conflict, resource busy, resource unavailable.

Error Code 2
The parameter Error Code 2 is user specific.

6.2.1.4.3 Data transport

This service is used to write and read the values of a single Combined Process Data object. This service may be used only via MS2 AR. Table 10 shows the parameters of the service.

With this service the data of a Process Data object are transferred

— from the DP-master to the DP-slave,
— from the DP-slave to the DP-master (Class 2), or
— from the DP-master to the DP-slave and back.

Table 10 – Data transport

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
Index	M	M(=)		
Length	M	M(=)		
Data	C	C(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Length			M	M(=)
Data			C	C(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error Code 1			M	M(=)
Error Code 2			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired slot (typically a module).

Index
The parameter Index is used in the destination device to address the desired data.

Length
In the Request the parameter Length indicates the number of octets of the parameter "Data" that are transferred from the Client to the Server. In the Response the parameter

Length indicates the number of octets of the parameter "Data" that are transferred from the Server to the Client.

Data
The parameter Data contains the values transferred from the Client to the Server and vice versa and consists of the number of octets indicated in the parameter Length. This parameter shall be composed of the data types defined in IEC 61158-5-10, Clause 5.

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

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

Error Decode
This parameter selects one of the following Error schemes: DPV1, FMS, Profile_Specific.

Error Code 1
In case of Error Decode equal DPV1 the Error Code 1 assumes one of the following values:
read error, write error, module failure, version conflict, feature not supported, user specific, invalid index, write length error, invalid slot, type conflict, invalid area, state conflict, access denied, invalid range, invalid parameter, invalid type, read constrain conflict, write constrain conflict, resource busy, resource unavailable.

Error Code 2
The parameter Error Code 2 is user specific.

6.2.2 I/O data ASE

6.2.2.1 Overview

In the DP Application Layer environment, the default application process of a DP-slave may contain I/O Data objects that shall be cyclically conveyed over the network. Therefore the I/O Data ASE defines attributes of the I/O Data objects and provides a set of services used to get and set the buffer to transport their values. The I/O Data objects are separated in an Input Data object and an Output Data object.

An Input Data object or an Output Data object shall be partitioned in Input Data or Output Data Elements which represent the I/O Data of the modules. The remote Input Data subscribed to in combination with the Output Data represent the real output data of the DP-slave.

Additional services are provided to read acyclically the values of the I/O Data objects and to indicate new values for the Input Data object and the Output Data object. Furthermore services are provided to indicate the presence and to get the content of new subscribed Input Data.

The I/O Data objects are implicitly addressed through the related services. The granularity of input or output data in a Server/Publisher is according to the correspondent MS0 Configuration Elements object. A Client is always related to the whole Input Data and/or Output Data of a Server. A Subscriber may be related to the whole Input Data or a part of the Input Data of a Publisher according to the attribute values of the DXB-Linktable object.

The I/O Data ASE uses the Client/Server and Publisher/Subscriber access model. The Client/Server model is characterized by a Client application conveying the value for the Output Data object to the Server's buffer. The Server application fetches this value by a Get Output service. The receipt of a new value is indicated by the New Output service. The Server application conveys the value of it's Input Data object via the Set Input service to the Server's buffer as soon as one of the real object values changes. The Client application gets this value

by the Get Input service. The transmission of the Server buffer to the Client buffer and vice versa is done asynchronously to the Get Output and Set Input services.

The Publisher/Subscriber model is characterized by DP-slaves acting as Publisher which publish the values of the Input Data objects and Subscribers which subscribe one or all values of the Input Data object of one or several Publisher (DP-slaves). The Publisher application conveys the value of it's Input Data object (Simple or Extended Input Data object) via the Set Input service to the Server's buffer as soon as one of the real object values changes. The Client (DP-master (Class 1)) causes the Publisher to publish the values of the Input Data object so that these values are available at the Client side and at every Subscriber on the network. The Subscriber subscribes to as many Input Data objects as the attribute values of the DXB-Linktable or DXB-Subscribertable indicate. The receipt of new values of subscribed Input Data are indicated by the New Publisher Data service at the Subscriber. The Subscriber application gets these values by the Get Publisher Data service.

It is possible to update only parts of the values of the I/O Data objects according to the given granularity regarding the structure of the objects. Furthermore, it is also possible that a DP-slave has only either an Input Data object or an Output Data object.

Most DP-slaves support only Octet or Word structures for the Input Data object and the Output Data object. This type of devices use the Simple Input and Output Data object. However, complex devices use Floating Point values or Doubleword structures or even combinations of several data types. The extended Input/Output Data object reflect the requirements of these enhanced devices.

The values of the I/O Data objects of all or a group of DP-slaves within a given network may be synchronized by its assigned DP-master with help of the Global Control service.

The synchronization of the values of the Input Data objects is achieved by the Freeze Command and the synchronization of the values of the Output Data objects is achieved by the Sync Command. This synchronization mechanism has no effect on the reception of the subscribed Input Data.

With each Freeze Command the current value of the Input Data object in the MS0 buffer is frozen in the addressed DP-slaves. If the Server application now updates the value of the Input Data object by the Set Input service then this value will not be issued to the buffer until the next Freeze Command occurs. That means, that the addressed DP-slaves will convey the "frozen" values of the Input Data objects. The synchronization of the Input Data objects will be switched off by a Freeze Command with the value Unfreeze.

With the first Sync Command the current value of the Output Data object in the MS0 buffer is frozen in the addressed DP-slaves. With each further Sync Command the previously transmitted values of the Output Data objects will be updated to the MS0 AR buffer for access with the Get Output service. The synchronization of the Output Data objects will be switched off by a Sync Command with the value Unsync. A detailed description is given in 6.2.8.3.2.

The values of the subscribed Input Data objects will not be synchronized with the values of the Output Data object in the Subscriber.

The DP-master (Class 1) sends information about its own operation mode to its assigned DP-slaves for synchronization. This will be done at every change of the operation mode of the DP-master (Class 1) or occurs in a time interval which can be parameterized. If the DP-master's state is CLEAR then the values of the Output Data object shall be set to a value corresponding to a safe state.

The values of the I/O Data objects of all or a group of DP-slaves within a given network may be synchronized by the assigned DP-master (Class 1) by a special synchronization mechanism (Isochronous Mode).

The synchronization of the values of the Input Data objects and the synchronization of the values of the Output Data objects is achieved by the synchronization message (SYNCH) and fixed time settings related to the synchronization message within the involved DP-slaves and the DP-master (Class 1).

The values of the subscribed Input Data objects in the Subscriber supporting the Isochronous Mode may also be synchronized by the Isochronous Mode.

The DP-master (Class 1) communicates via the MS0 AR with all Input Data and Output Data objects of the assigned DP-slaves simultaneously. Additionally the Client application may communicate with an Input Data object, an Output Data object, an Input Data Element or an Output Data Element of a single DP-slave.

The DP-master (Class 2) communicates via the MS0 AR with the Input Data and Output Data object of a single DP-slave.

A DP-slave, configured as Subscriber, communicates with the Input Data object(s) of one or more DP-slaves, configured as Publisher and to which he is subscribed to via the MS0 AR.

The mapping of the subscribed Input Data to all or parts of the Output Data Elements of the DP-slave may be done application specific or may be accomplished by the application of the DP-slave with the DXB-Subscribertable in conjunction with the configuration data.

The formal model of the I/O Data ASE is presented next, followed by a description of its services.

6.2.2.2 I/O data class specification

6.2.2.2.1 Simple input data class specification

6.2.2.2.1.1 Template

The Simple Input Data object is described by the following template:

DP ASE: **I/O Data ASE**
CLASS: **Simple Input Data**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of Input Data Elements
2.1.	(m)	Attribute:	Slot Number
2.2.	(m)	Attribute:	Simple Input Data Description
2.2.1.	(m)	Attribute:	Length
2.2.2.	(m)	Attribute:	Format
2.2.3.	(m)	Attribute:	Consistency
2.2.4.	(o)	Attribute:	Name

SERVICES:

1.	(m)	OpsService:	Set Input
2.	(o)	OpsService:	Read Input
3.	(m)	OpsService:	Get Input
4.	(o)	OpsService:	New Input
5.	(m)	OpsService:	Get Publisher Data
6.	(o)	OpsService:	New Publisher Data
7.	(m)	OpsService:	Global Control
8.	(o)	OpsService:	SYNCH
9.	(o)	OpsService:	SYNCH Delayed
10.	(o)	OpsService:	DX Finished
11.	(o)	OpsService:	SYNCH Event

6.2.2.2.1.2 Attributes

Implicit
The attribute Implicit indicates that the Simple Input Data object is implicitly addressed by the services.

List of Input Data Elements
One Input Data Element is composed of the following list elements:

Slot Number
This attribute defines to which module the Input Data Element belongs.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Simple Input Data Description
This attribute consists of the following attributes.

Length
The attribute Length defines the number of octets of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Unsigned8

Allowed values: 1 to 128

Format
This attribute defines the format of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 11.

Table 11 – Format (simple input data description)

Value	Meaning
0	Octet
1	Word

Consistency
This attribute defines the Consistency of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 12.

Table 12 – Consistency (simple input data description)

Value	Meaning
0	Consistency over the defined Format of the Input Data Element
1	Consistency over the whole Input Data Element

Name
This attribute defines the name of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Visible String(32)

6.2.2.2.1.3 Invocation of the simple input data object

For the invocation of the Simple Input Data object the following rules apply:

— Simple Input Data objects shall not exceed the total length of 244 octets. The attributes List of Input Data Elements and Length have to be set accordingly.

— Only one Simple Input Data object shall be invoked in a DP-slave.

— A Simple Input Data object shall only be invoked if no Extended Input Data object (see 6.2.2.2.3) has been invoked.

6.2.2.2.2 Simple output data class specification

6.2.2.2.2.1 Template

The Simple Output Data object is described by the following template:

DP ASE:			**I/O Data ASE**
CLASS:		**Simple Output Data**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of Output Data Elements
2.1.	(m)	Attribute:	Slot Number
2.2.	(m)	Attribute:	Simple Output Data Description
2.2.1.	(m)	Attribute:	Length
2.2.2.	(m)	Attribute:	Format
2.2.3.	(m)	Attribute:	Consistency
2.2.4.	(o)	Attribute:	Name
SERVICES:			
1.	(m)	OpsService:	Set Output
2.	(o)	OpsService:	Read Output
3.	(m)	OpsService:	Get Output
4.	(o)	OpsService:	New Output
5.	(m)	OpsService:	Global Control
6.	(o)	OpsService:	SYNCH
7.	(o)	OpsService:	SYNCH Delayed
8.	(o)	OpsService:	DX Finished
9.	(o)	OpsService:	SYNCH Event

6.2.2.2.2.2 Attributes

Implicit
The attribute Implicit indicates that the Simple Output Data object is implicitly addressed by the services.

List of Output Data Elements
One Output Data Element is composed of the following list elements:

Slot Number
This attribute defines to which module the Output Data Element belongs.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Simple Output Data Description
This attribute consists of the following attributes.

Length
The attribute Length defines the number of octets of the Output Data Element. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Unsigned8

Allowed values: 1 to 128

Format
This attribute defines the format of the Output Data Element. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 13.

Table 13 – Format (simple output data)

Value	Meaning
0	Octet
1	Word

Consistency
This attribute defines the Consistency of the Output Data Element. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 14.

Table 14 – Consistency (simple output data)

Value	Meaning
0	Consistency over the defined Format of the Output Data Element
1	Consistency over the whole Output Data Element

Name
This optional attribute defines the name of the Output Data Element.

Attribute Type: Visible String(32)

6.2.2.2.2.3 Invocation of the simple output data object

For the invocation of the Simple Output Data object the following rules apply:

— Simple Output Data objects shall not exceed the total length of 244 octets. The attributes List of Output Data Elements and Length have to be set accordingly.

— Only one Simple Output Data object shall be invoked in a DP-slave.

— A Simple Output Data object shall only be invoked if no Extended Output Data object (see 6.2.2.2.4) has been invoked.

6.2.2.2.3 Extended input data class specification

6.2.2.2.3.1 Template

The Extended Input Data object is described by the following template:

DP ASE: **I/O Data ASE**
CLASS: **Extended Input Data**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of Input Data Elements
2.1.	(m)	Attribute:	Slot Number
2.2.	(m)	Attribute:	Selector of Data Type
2.2.1.	(s)	Attribute:	Extended Input Data Description
2.2.2.	(s)	Attribute:	Simple Input Data Description
2.2.2.1.	(m)	Attribute:	Length
2.2.2.2.	(m)	Attribute:	Format
2.2.2.3.	(m)	Attribute:	Consistency
2.3.	(o)	Attribute:	Name

SERVICES:

1.	(m)	OpsService:	Set Input
2.	(o)	OpsService:	Read Input
3.	(m)	OpsService:	Get Input
4.	(o)	OpsService:	New Input
5.	(m)	OpsService:	Get Publisher Data
6.	(o)	OpsService:	New Publisher Data
7.	(m)	OpsService:	Global Control
8.	(o)	OpsService:	SYNCH
9.	(o)	OpsService:	SYNCH Delayed
10.	(o)	OpsService:	DX Finished
11.	(o)	OpsService:	SYNCH Event

6.2.2.2.3.2 Attributes

Implicit

The attribute Implicit indicates that the Extended Input Data object is implicitly addressed by the services.

List of Input Data Elements
One Input Data Element is composed of the following list elements:

Slot Number
This attribute defines to which module the Input Data Element belongs.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Selector of Data Type
This attribute defines whether this module is described according to the Extended Input Data Description or according to the Simple Input Data Description. In case of an Extended Input Data Description, consistency is over the whole Input Data Element.

Extended Input Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Simple Input Data Description
This attribute consists of the following attributes.

Length
The attribute Length defines the number of octets of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Unsigned8

Allowed values: 1 to 128

Format
This attribute defines the format of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 15.

Table 15 – Format (extended input data)

Value	Meaning
0	Octet
1	Word

Consistency
This attribute defines the Consistency of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 16.

Table 16 – Consistency (extended input data)

Value	Meaning
0	Consistency over the defined Format of the Input Data Element
1	Consistency over the whole Input Data Element

Name
This optional attribute defines the name of the Input Data Element. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Visible String(32)

6.2.2.2.3.3 Invocation of the extended input data object

For the invocation of the Extended Input Data object the following rules apply:

— Extended Input Data objects shall not exceed the total length of 244 octets. The attributes List of Input Data Elements, Input Data Description respectively Format and Length have to be set accordingly.

— Only one Extended Input Data object shall be invoked in a DP-slave.

— An Extended Input Data object shall only be invoked if no Simple Input Data object has been invoked.

6.2.2.2.4 Extended output data class specification

The Extended Output Data object is described by the following template:

DP ASE: **I/O Data ASE**
CLASS: **Extended Output Data**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of Output Data Elements
2.1.	(m)	Attribute:	Slot Number
2.2.	(m)	Attribute:	Selector of Data Type
2.2.1.	(s)	Attribute:	Extended Output Data Description
2.2.2.	(s)	Attribute:	Simple Output Data Description
2.2.2.1.	(m)	Attribute:	Length
2.2.2.2.	(m)	Attribute:	Format
2.2.2.3.	(m)	Attribute:	Consistency
2.3.	(o)	Attribute:	Name

SERVICES:

1.	(m)	OpsService:	Set Output
2.	(o)	OpsService:	Read Output
3.	(m)	OpsService:	Get Output
4.	(o)	OpsService:	New Output
5.	(m)	OpsService:	Global Control
6.	(o)	OpsService:	SYNCH
7.	(o)	OpsService:	SYNCH Delayed
8.	(o)	OpsService:	DX Finished
9.	(o)	OpsService:	SYNCH Event

6.2.2.2.4.1 Attributes

Implicit
The attribute Implicit indicates that the Extended Output Data object is implicitly addressed by the services.

List of Output Data Elements
One Output Data Element is composed of the following list elements:

Slot Number
This attribute defines to which module the Output Data Element belongs.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Selector of Data Type
This attribute defines whether this module is described according to the Extended Output Data Description or according to the Simple Output Data Description. In case of an Extended Output Data Description, consistency is over the whole Output Data Element.

Extended Output Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Simple Output Data Description
This attribute consists of the following attributes.

Length
The attribute Length defines the number of octets of the Output Data Element. The value of this attribute is determined by the corresponding configuration data. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Unsigned8

Allowed values: 1 to 128

Format
This attribute defines the format of the Output Data Element. The value of this attribute is determined by the corresponding configuration data. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 17.

Table 17 – Format (extended output data)

Value	Meaning
0	Octet
1	Word

Consistency
This attribute defines the Consistency of the Output Data Element. The value of this attribute is determined by the corresponding configuration data. The value of this attribute is determined by the corresponding configuration data.

The allowed values are shown in Table 18.

Table 18 – Consistency (extended output data)

Value	Meaning
0	Consistency over the defined Format of the Output Data Element
1	Consistency over the whole Output Data Element

Name
This optional attribute defines the name of the Output Data Element. The value of this attribute is determined by the corresponding configuration data. The value of this attribute is determined by the corresponding configuration data.

Attribute Type: Visible String(32)

6.2.2.2.4.2 Invocation of the extended output data object

For the invocation of the Extended Output Data object the following rules apply:

— Extended Output Data objects shall not exceed the total length of 244 octets. The attributes List of Output Data Elements, Output Data Description respectively Format and Length have to be set accordingly.

— Only one Extended Output Data object shall be invoked in a DP-slave.

— An Extended Output Data object shall only be invoked if no Simple Output Data object has been invoked.

6.2.2.3 I/O data service specification

6.2.2.3.1 Set input

This service shall be used by a DP-slave to pass the value of a Simple Input Data object or an Extended Input Data object to the DP Application Layer. This service shall only be used in conjunction with the MS0 AR.

By this service the application process of the DP-slave updates the local buffer. According to the cyclic buffer to buffer transportation characteristics of the used application relationship the following behavior is possible:

— The Set Input requests come faster than the contents of the buffers are transported over the network to the Client and Subscribers. In this case not each value of the Input Data object provided with the Set Input service is conveyed over the network. Only the latest value is transported, others are locally overwritten.

— The Set Input requests are issued slower than the contents of the buffers are transported over the network to the Client and Subscribers. In this case each value of the Input Data object provided with the Set Input service is conveyed more than once over the network.

— If the Set Input requests are issued synchronous with the transport of the contents of the buffers each value of the Input Data object provided with the Set Input service is conveyed over the network.

Table 19 shows the parameters of the service.

Table 19 – Set Input

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Input Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Input Data
This parameter contains the value of the Input Data object.

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

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

6.2.2.3.2 Read input

By this service any DP-master (Class 2) can read the value of the Input Data object of a DP-slave. The service will only deliver a positive result if the DP-slave is already in the data exchange mode with its assigned DP-master (Class 1). This service shall only be used in conjunction with the MS0 AR. Table 20 shows the parameters of the service.

Table 20 – Read Input

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result(+)		S
AREP		M
Input Data		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

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

Input Data
This parameter contains the requested value of the Input Data object.

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

Status
This parameter provides detailed information about the occurred error.

Allowed values: DS, NA, RS, UE, NR, RE

6.2.2.3.3 Get input

This service shall only be used in conjunction with the MS0 AR. It is used by a DP-master (Class 1) to read:

— the values of all Input Data objects of the assigned DP-slaves if the parameter CREP and Slot Number have not been used in the Get Input service request

— the value of the Input Data object of a single assigned DP-slave if the parameter Slot Number has not been used in the Get Input service request

— the value of an Input Data Element of a single assigned DP-slave if the parameter CREP and Slot Number have been used in the Get Input service request

This service may be acknowledged negatively if the DP-master (Class 1) operates in the Isochronous Mode in the following cases:

— if the service request is made not within the allowed window of interactions of the Enhanced Synchronized Isochronous Mode (SE)

— if the service request is made between the service Set Output with the parameter Final set to TRUE and the next SYNCH indication in case of Buffered Synchronized Isochronous Mode (SE)

— if the addressed DP-slave is deactivated (SC)

Table 21 shows the parameters of the service.

Table 21 – Get Input

Parameter name	Req	Cnf
Argument	M	
AREP	M	
CREP	U	
Slot Number	U	
Result(+)		S
AREP		M
CREP		C
Slot Number		C
Input Data		M
Result(-)		S
AREP		M
CREP		C
Slot Number		C
Status		C

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
If this parameter is present then the Input Data object of a single DP-slave is addressed.

Slot Number
If this parameter is present then an Input Data Element of a single DP-slave is addressed. If this parameter is used the parameter CREP shall also be present.

Allowed values: 1 to 244

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

Input Data
This parameter contains the selected value.

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

Status
This parameter provides detailed information about the occurred error.

Allowed values: SC, SE

6.2.2.3.4 New input

The New Input service signals to the application of the DP-master (Class 1) that new values of all Input Data objects of a DP-slaves which are related to the MS0 AR are present. This service shall only be used in conjunction with the MS0 AR. Table 22 shows the parameters of the service.

Table 22 – New Input

Parameter name	Ind
Argument	M
AREP	M
CREP	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier of the AR which passed the indication.

CREP
This parameter is the local identifier of a single DP-slave.

6.2.2.3.5 Set output

By this service the Client application process of the DP-master (Class 1) requests the conveyance of:

- the values of all Output Data objects of the assigned DP-slaves if the parameter CREP and Slot Number have not been used in the Set Output service request
- the value of the Output Data object of a single assigned DP-slave if the parameter Slot Number has not been used in the Set Output service request

the value of an Output Data Element of a single assigned DP-slave if the parameter CREP and Slot Number have been used in the Set Output service request

This service shall only be used in conjunction with the MS0 AR.

According to the cyclic buffer to buffer transportation characteristics the following behavior is possible:

- The Set Output requests come faster than the contents of the buffers are transported over the network. In this case only the values provided with the latest Set Output service are conveyed over the network.
- The Set Output requests issued slower than the contents of the buffers are transported over the network. In this case the values provided with each Set Output service are conveyed more than once over the network.

If the Set Output requests are issued synchronous with the transport of the contents of the buffers the values provided with each Set Output service are conveyed once over the network.

This service may be acknowledged negatively if the DP-master (Class 1) operates in the Isochronous Mode in the following cases:

- if the service request is made not within the allowed window of interactions of the Enhanced Synchronized Isochronous Mode (SE)
- if the service request is made between the service Set Output with the parameter Final set to TRUE and the next SYNCH indication in case of Buffered Synchronized Isochronous Mode (SE)

if the addressed DP-slave is deactivated (SC)

Table 23 shows the parameters of the service.

Table 23 – Set Output

Parameter name	Req	Cnf
Argument	M	
AREP	M	
CREP	U	
Slot Number	U	
Output Data	M	
Final	M	
Result(+)		S
AREP		M
CREP		C
Slot Number		C
Result(-)		S
AREP		M
CREP		C
Slot Number		C
Status		C

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
If this parameter is present then the Output Data object of a single DP-slave is addressed.

Slot Number
If this parameter is present then an Output Data Element of a single DP-slave is addressed. If this parameter is used the parameter CREP shall also be present.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Output Data
This parameter contains the selected value.

Final
This parameter indicates to the DP Application Layer to change the intermediate buffer of the Input Data and of the Output Data of the DP-slaves. The DP Application Layer changes these buffers after the DX Finished indication and before the next SYNCH indication of the Isochronous DP cycle in case of Buffered Synchronized Isochronous Mode.

The allowed values are shown in Table 24.

Table 24 – Final

Value	Meaning
TRUE	change of the intermediate buffer of the Input Data and of the Output Data shall be done
FALSE	change of the intermediate buffer of the Input Data and of the Output Data is not requested

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

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

Status
This parameter provides detailed information about the occurred error.

Allowed values: SC, SE

6.2.2.3.6 Read output

By this service any DP-master (Class 2) can read acyclic the value of the Output Data object of a DP-slave. The service will only deliver a positive result if the DP-slave is already in the data exchange mode with its assigned DP-master (Class 1). This service shall only be used in conjunction with the MS0 AR.

If the DP-master (Class 1) changes the operation mode to CLEAR the DP-slave shall set the value of the Output Data object to a value corresponding to a safe state (e.g. zero, last or default value). In this case the value received with this service may not correspond to the real value of the Output Data object.

Table 25 shows the parameters of the service.

Table 25 – Read Output

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result(+)		S
AREP		M
Output Data		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

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

Output Data
This parameter contains the requested value of the Output Data object.

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

Status
This parameter provides detailed information about the occurred error.

Allowed values: DS, NA, RS, UE, NR, RE

6.2.2.3.7 Get output

This service shall be used by a DP-slave to read the value for its Simple Output Data object or Extended Output Data object from the DP Application Layer. This service shall only be used in conjunction with the MS0 AR.

Table 26 shows the parameters of the service.

Table 26 – Get Output

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result(+)		S
AREP		M
Output Data		M
Clear Flag		M
New Flag		M
Result(-)		S
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

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

Output Data
This parameter contains the value for the Output Data object.

Clear Flag
This parameter indicates whether the value for the Output Data object is valid or cleared.

The allowed values are shown in Table 27.

Table 27 – Clear Flag

Value	Meaning
0	value for the Output Data object valid (DP-master in the state OPERATE)
1	value for the Output Data cleared (DP-master in the state CLEAR)

New Flag
This parameter indicates whether a new value for the Output Data object is received or not.

The allowed values are shown in Table 28.

Table 28 – New Flag

Value	Meaning
0	no new value received
1	new value received

Result(-)

This selection type parameter indicates that the service request failed.

6.2.2.3.8 New output

This service indicates to the application of the DP-slave that a new value for its Simple Output Data object or Extended Output Data object is present. Table 29 shows the parameters of the service.

Table 29 – New Output

Parameter name	Ind
Argument	M
AREP	M
Clear Flag	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier of the AR which passed the indication.

Clear Flag
This parameter indicates whether the value for the Output Data object is valid or cleared.

The allowed values are shown in Table 30.

Table 30 – Clear Flag

Value	Meaning
0	value for the Output Data object valid (DP-master in the state OPERATE)
1	value for the Output Data cleared (DP-master in the state CLEAR)

6.2.2.3.9 Global control

This service is used by the DP-master (Class 1) for the synchronization of the values of the Input Data objects (Freeze Command) and the synchronization of the values of the Output Data objects (Sync Command) of a group or all assigned DP-slaves. This service shall only be used in conjunction with the MS0 AR.

If the DP-master (Class 1) operates in the Isochronous Mode with all or a group of its assigned DP-slaves group 8 of the parameter Group Select of the Global Control service is no longer a valid group value for this parameter.

Furthermore this service is used by the DP-master (Class 1) to send information about its own operation mode (Clear Command) to all assigned DP-slaves.

Table 31 shows the parameters of the service.

Table 31 – Global Control

Parameter name	Req	Ind	Cnf
Argument	M	M(=)	
AREP	M	M(=)	
Clear Command		M	
Sync Command	M	M(=)	
Freeze Command	M	M(=)	
Group Select	M	M(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Clear Command
This parameter contains the Clear Command.

The allowed values ar shown in Table 32.

Table 32 – Clear Command

Value	Meaning
0	DP-master (Class 1) in the state OPERATE: The values for the Output Data objects are valid.
1	DP-master (Class 1) in the state CLEAR: The values for the Output Data objects shall be set to a value corresponding to a safe state.

Sync Command
This parameter controls the Sync Operation.

The allowed values shown in Table 33.

Table 33 – Sync Command

Value	Meaning
0	no action
1	Sync
2	Unsync

Freeze Command
This parameter controls the Freeze Operation.

The allowed values are shown in Table 34.

Table 34 – Freeze Command

Value	Meaning
0	no action
1	Freeze
2	Unfreeze

Group Select
This parameter determines which group(s) of assigned DP-slaves shall be addressed. The control command takes effect if the AND operation between the attribute Group Identifier of the ARL object of the DP-slave and the Group Select parameter results in a value different from zero. If the parameter Group Select is zero all assigned DP-slaves are addressed.

If the DP-master (Class 1) operates in the Isochronous Mode group 8 is not allowed for this parameter. The service request shall be confirmed negatively with an appropriate status value (GE).

Result(+)
This selection type parameter indicates that the service request has been transmitted.

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

Status
This parameter provides detailed information about the occurred error.

Allowed values: DS, NO, IV, GE

6.2.2.3.10 New publisher data

This service indicates to the application of the DP-slave that a new value from a subscribed Input Data object is present. Table 35 shows the parameters of the service.

Table 35 – New publisher data

Parameter name	Ind
Argument	M
AREP	M
CREP	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier for the desired AR.

CREP
This parameter is the local identifier of the CR (Publisher) which passed the indication.

6.2.2.3.11 Get publisher data

This service shall be used by the DP-slave (Subscriber) application to read the value from a particular subscribed Input Data object from the DP Application Layer. This service shall only be used in conjunction with the MS0 AR.

Table 36 shows the parameters of the service.

Table 36 – Get publisher data

Parameter name	Req	Cnf
Argument	M	
AREP	M	
CREP	M	
Result(+)		S
AREP		M
CREP		M
Data		M
New Flag		M
Result(-)		S
AREP		M
CREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
This parameter is the local identifier for the desired CR.

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

Data
This parameter contains the value from the subscribed Input Data object.

New Flag
This parameter indicates whether a new value from the subscribed Input Data object is received or not.

The allowed values are shown in Table 37.

Table 37 – New Flag

Value	Meaning
0	no new value received
1	new value received

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

6.2.2.3.12 SYNCH

The service SYNCH indicates to the application of the DP-master (Class 1) that a new Isochronous DP cycle has been started.

Table 38 shows the parameters of the service.

Table 38 – SYNCH

Parameter name	Ind
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier of the AR which passed the indication.

6.2.2.3.13 SYNCH delayed

The service SYNCH Delayed shall be passed to the application of the DP-master (Class 1) only if the last synchronization message was not sent within the allowed time $maxT_{SH}$. The value of the time $maxT_{SH}$ is contained in the related MS0 AR entry of the ARL of the DP-master (Class 1). The service also delivers the time difference measured according to the following rules:

- the synchronization message was sent earlier than $|T_{DP} - maxT_{SH}|$
- the synchronization message was sent later than $|T_{DP} + maxT_{SH}|$

Table 39 shows the parameters of the service.

Table 39 – SYNCH Delayed

Parameter name	Ind
Argument	M
AREP	M
Shift	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier of the AR which passed the indication.

Shift
This parameter contains the value of the measured time difference (shift) in units of t_{BIT}.

Attribute Type: Unsigned32

Allowed values: 1 to $2^{32}-1$

6.2.2.3.14 DX finished

The service DX Finished indicates to the application of the DP-master (Class 1) that the cyclic part (DX) of the Isochronous DP cycle is completed.

Table 40 shows the parameters of the service.

Table 40 – DX Finished

Parameter name	Ind
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier of the AR which passed the indication.

6.2.2.3.15 SYNCH event

The service SYNCH Event indicates to the application of the DP-slave if

- the DP-master (Class 1) and the DP Application Layer of the DP-slave has started to operate in the Isochronous Mode
- that a new Isochronous DP cycle has been started

the DP-master (Class 1) and the DP Application Layer of the DP-slave have stopped to operate in the Isochronous Mode.

Table 41 shows the parameters of the service.

Table 41 – SYNCH Event

Parameter name	Ind
Argument	M
AREP	M
Status	M

Argument
The argument shall convey the service specific parameters of the service indication.

AREP
This parameter is the local identifier of the AR which passed the indication.

Status
This parameter indicates the status of the Isochronous Mode.

The allowed values are shown in Table 42.

Table 42 – Status

Value	Meaning
IsoM Start	start of Isochronous Mode operation
IsoM SYNCH	start of a new Isochronous DP cycle
IsoM Stop	stop of Isochronous Mode operation

6.2.2.4 Behavior of I/O data objects

6.2.2.4.1 General behavior of the output data object

For performance reasons, the value for the Output Data object should be transferred on receipt as fast as possible to the object. The speed and the jitters of this action is a performance parameter of a DP-slave. This parameter determines the synchronization accuracy.

For highly synchronized DP-slaves, this action should be executed always at the same moment of time after the Sync Command.

6.2.2.4.2 Characteristics of a DP system with isochronous mode functionality

6.2.2.4.2.1 Basic behavior of input and output data in the isochronous mode

A DP system operating in the Isochronous Mode provides the following features (specified at data rates above 1 000 kbit/s and isochronous DP cycles between 500 µs and 32 ms)

- all DP-Devices taking part in the Isochronous Mode functionality synchronies their internal clock system with the synchronization message SYNCH with the Isochronous DP cycle
- all DP-Devices taking part in the Isochronous Mode functionality have to be on the same network (Isochronous Mode functionality over multiple networks is not possible; repeaters are allowed)
- the jitter of SYNCH-Event.ind is less than 1µs (Jitters within DP-slave is assumed as 0)
- in each DP cycle optionally an acyclic service of full length on the MS1 AR may be included (request or response) and be accomplished if the there is enough time to process the service completely before the next Isochronous DP cycle has to be started

the operation of a DP-master (Class 2) on the same network for configuration or diagnosis purposes is possible (the time to execute those services should be included in the cycle time).

In a DP system with enabled Isochronous Mode functionality DP-slaves with basic functionality can be used in conjunction with synchronized DP-slaves. The Isochronous Mode functionality tolerates and compensates communication errors like:

- faulty synchronization messages (SYNCH); the synchronization message may be disturbed at the DP-master (Class 1) at the generation or at the conveyance to the DP-slave or at the recognition at the DP-slave
- the Isochronous Mode is capable to compensate the jitters of the synchronization message produced by the DP-master (Class 1) and if present by network devices like repeaters; this jitter shall be less than 1µs (this includes the jitters within the DP-master (Class 1))
- the deviation of the real Isochronous DP cycle to the ideal, not disturbed Isochronous DP cycle shall not exceed 0,03%
- the SYNCH-Event.ind may be missed due to disturbed telegram within the Isochronous DP; up to three subsequently not in time sent or missing synchronization messages are allowed for a DP-master (Class 1) operating in the Isochronous Mode

due to disturbances the synchronisation messages may be destroyed at the conveyance to the DP-slave; the clock recovery mechanism in the DP-slave operating in the Isochronous Mode tolerates up to five subsequently not received synchronisation messages

If a DP system shall operate in the Isochronous Mode the following definitions shall apply.

— The DP-master (Class 1) and the correspondent DP-slaves shall support the Fail Safe mechanism.

— The DP-slave shall operate in the Isochronous Mode only if the parameter Fail Safe and IsoM Req provided by the parameterization are set to TRUE. Additionally the necessary IsoM Parameter object shall be provided with the Check User Prm or Check Ext User Prm service to the application of the DP-slave to set the Isochronous Mode parameter of the DP-slave.

6.2.2.4.2.2 Telegram sequences of a DP system operating in the isochronous mode

Figure 15 shows the sequences of DP systems operating in the Isochronous Mode. A DP cycle in the Isochronous context is considered from the SYNCH (synchronization message) to the next SYNCH.

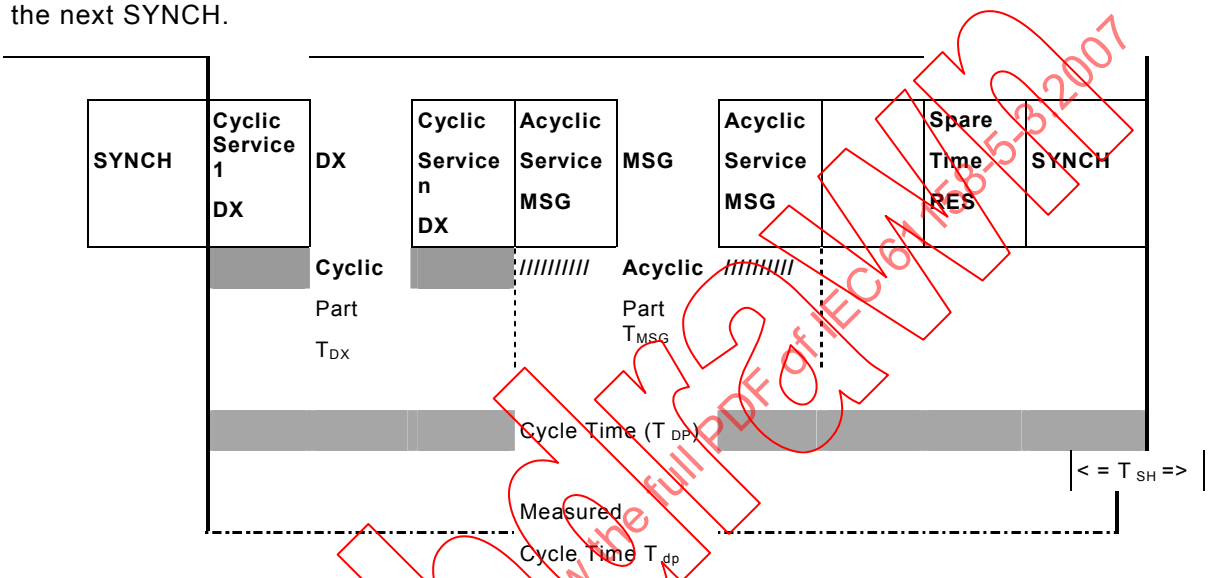

Figure 15 – Sequence of an isochronous DP cycle with one DP-master (class 1)

For the Isochronous Mode the following definitions shall apply.

T_{DX} (Data eXchange Time)
The time T_{DX} may elapse to handle all MS0 communication of one Master (Class 1) of fieldbus DP.

T_{DP} (Data and Parameter exchange Time = Isochronous DP cycle time)
The time T_{DP} may elapse to handle all parts of the Isochronous DP cyclic. The Isochronous DP cycle consists of the following parts:

- SYNCH-Event
- cyclic services (DX)
- acyclic services (MSG)

spare time (RES)

T_{MSG} (Message Time)
The times T_{MSG} may elapse to handle all acyclic services on the MS1 AR (T_{MSG1}) and MS2 AR (T_{MSG2}) of fieldbus DP as well as all other DL-services with Service_class = low. These acyclic services shall be executed after the cyclic services. To ensure an Isochronous DP cycle this part shall be limited.

T_{SH} (Shift Time)
This time shall be calculated with $T_{measured\ cycle}$ — T_{DP}. The shift may be caused due to the following reasons:

- delayed sending of the synchronization message (SYNCH) at the DP-master (Class 1)

inexact calculation of the time T_{DP} at the DP-master (Class 1)

The maximum value of this time (maxT_{SH}) is implementation and configuration dependent. If the time T_{SH} exceeds a certain limit (maxT_{SH}) the application in the DP-master (Class 1) will be informed with the SYNCH Delayed service indicating the measured shift.

Jitters
Jitters may be caused due to the following reasons:

(This definition determines the required parts for a PLL- Minimising Jitters influence, capture drifts and compensate signal losses. It shall be used as a base for quality assurance)

- jitters caused by network components like repeaters or fibre optic interfaces
- transmission rate dependent jitters (jitters < 1 t_{BIT} for the sender, receiver and the network components between)

implementation dependent reaction time in the sending and receiving station and the network components between

The following changes of T_{DP} are not part of the Jitters definition

- maximum deviation of T_{DP} in the sending and receiving station (jitters < 0,03%·T_{DP})
- losses of clock signaling
- deviation changes due to temperature and ageing (1 ppm within the range of a minute)

SYNCH is sent too late

RES (Spare time)
This part is defined as the rest of the time between the cyclic part, the acyclic part and the Token handling and the next synchronization message. The duration of the spare time (RES) depends on the services sent in the actual Isochronous DP cycle.

6.2.2.4.2.3 Time relationships of a DP system operating in isochronous mode

Figure 16 shows the time relationships of the applications of the DP-master (Class 1) and 3 DP-slaves operating in the Isochronous Mode.

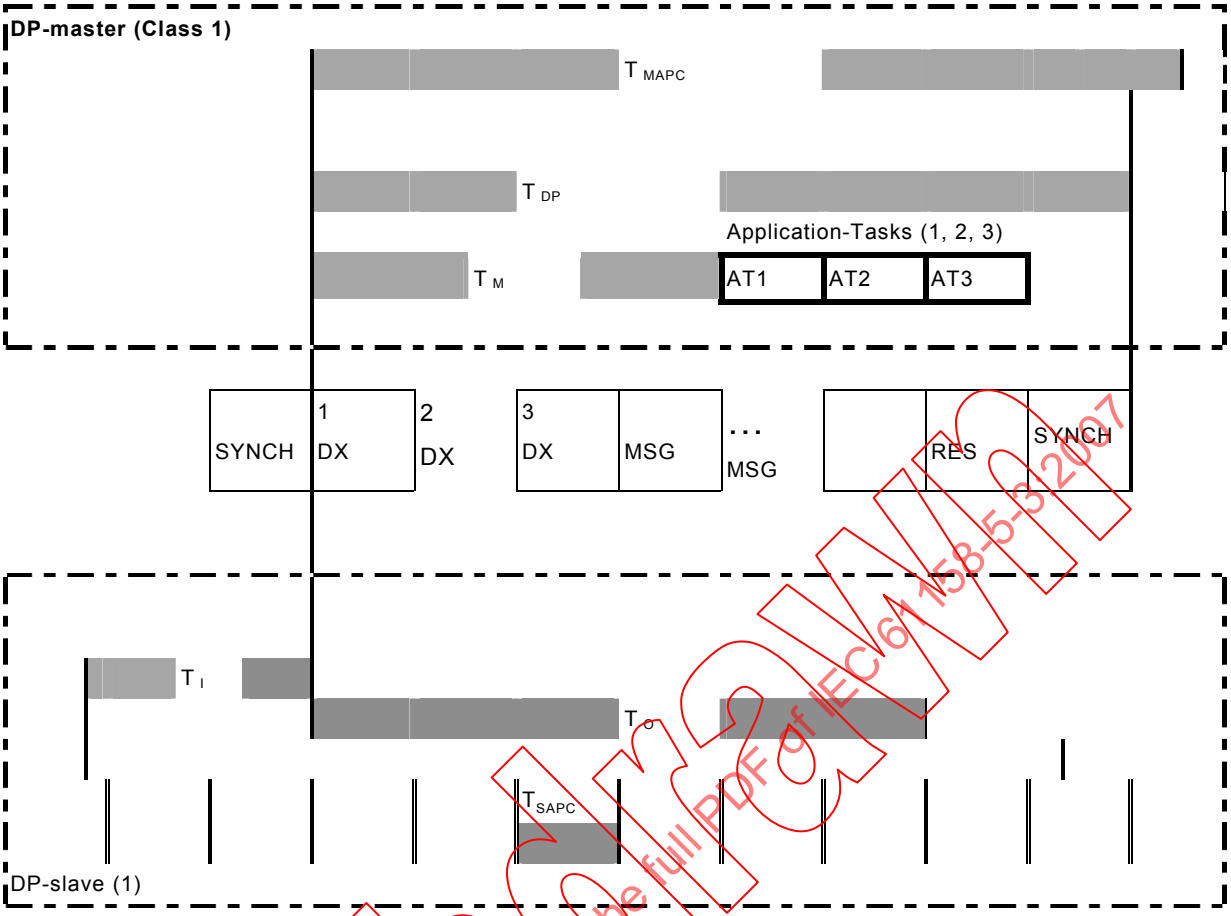

Figure 16 – Additional time relationships in a DP system operating in isochronous mode

For the Isochronous Mode the following time definitions for the application relationships shall be applied.

T_{I_MIN} (Minimum T_I)
This time parameter indicates the minimum time necessary to get and update the Input Data of an individual DP-slave.

T_I (Input Time)
This time is used to trigger the acquisition and update of the Input Data object in a DP-slave operating in Isochronous Mode and may be calculated by the configuration tool. This time is related to the end of the Isochronous DP cycle.

T_{O_MIN} (Minimum T_O)
This time parameter indicates the minimum time that is necessary at the end of the cyclic part of the Isochronous DP cycle (T_{DX}) to get and output the Output Data given in units of T_{BASE_IO} of an individual DP-slave.

T_O (Output Time)
This time is used to trigger the output processing of the Output Data object in a DP-slave operating in Isochronous Mode and may be calculated by the configuration tool. This time is related to the beginning of the Isochronous DP cycle.

T_M (Master Time)
This time is used to shift the processing of the application tasks in the DP-master (Class 1) behind the end of the cyclic part of the Isochronous DP cycle (T_{DX}). This parameter may be set by the application in the DP-master (Class 1) according to $T_M > T_{DX}$. If the time T_M is set to 0 (special case) the DP-master (Class 1) may process input data conveyed a complete Isochronous DP cycle before.

T_{MAPC}, T_{SAPC} (Master Application Cycle Time, Slave Application Cycle Time)
These parameters indicate the time the applications of the DP-master (Class 1) or of the DP-slave need to process an application task completely. The settings of these cycle time parameter may be done device specific. It is allowed to have different settings for the Master Application Cycle Time and Slave Application Cycle Time in each DP-slave using the common time base T_{DP}. The following rules shall apply:

— $T_{MAPC} \geq T_{DP} \geq T_{SAPC}$

— $T_{MAPC} = n \cdot T_{DP}$ (n = 1 to 14; limited by the slave life sign)

The Time relationships of a DP system, especially of the DP-slave, operating in the Isochronous Mode is described only for the so called compact DP-slave. To make the model also suitable for modular DP-slaves additional aspects have to be added. Especially the time relationships have to be extended and some additional time parameters may be necessary.

At least the following constraints have to be addressed:

— at a modular DP-slave the time parameters depend on the plugged in modules.

- each module may add also a time offset (each module type may have an individual time offset)
- the setting of T_I and T_O may be restricted.

the device itself may add an additional time offset due to the "transfer" time of the I/O data between the Interface device and the modules.

6.2.2.4.2.4 Example of a DP system with optimized isochronous DP cycle

In the following example the application task 1 and 2 in the DP-master (Class 1) may be processed in each Isochronous DP cycle completely.

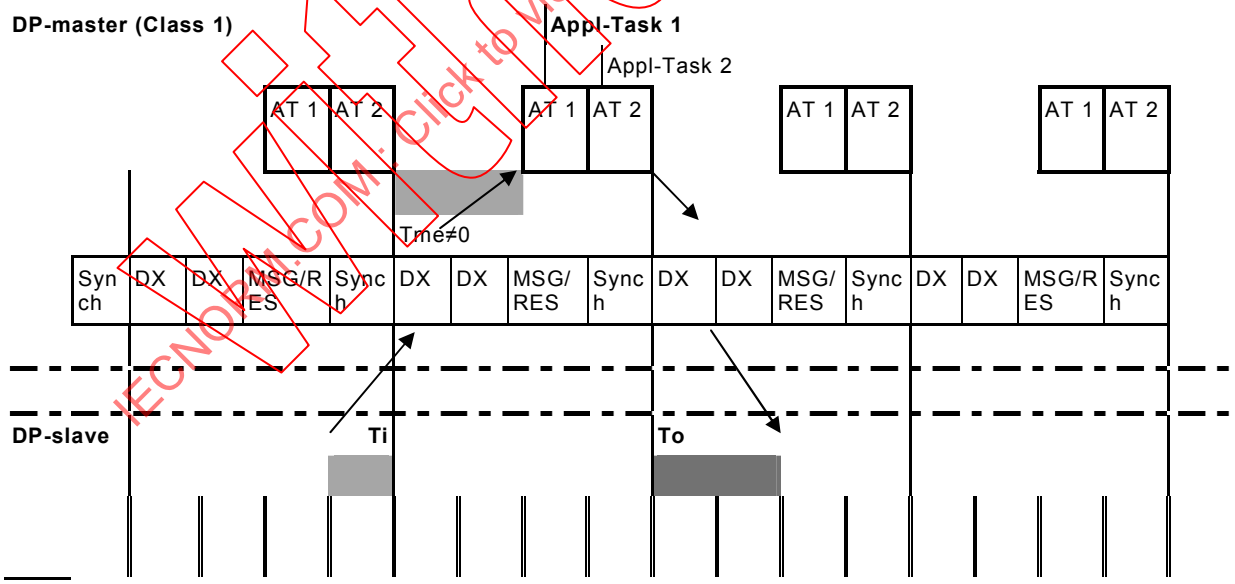

Figure 17 – DP system with optimized isochronous DP cycle

Figure 17 shows a DP System where the sequence of the acquisition and conveyance of the Input Data, processing of the application task and the transfer and output of the Output Data are optimized. This optimization reduces the necessary cycles to perform a complete control loop to less than 3 Isochronous DP cycles and is characterized by the following phases:

- Phase 1: acquisition and update of the value of the Input Data in the DP-slave
- Phase 2: conveyance of the value of the Input Data to the DP-master (Class 1) and processing of the application task 2 with the actual value of the Input Data and transfer of the new value for the Output Data from the application task 2 to the DP-master (Class 1)
- Phase 3: conveyance of the value for the Output Data to the DP-slave and output of the new Output Data

This model of synchronization requires enhanced processing capabilities of the DP-master (Class 1) and requires an optimized scheduling in the DP-master (Class 1) and in the DP-slaves.

The dead time in this model of synchronization is given with: $T_D = T_{DP} + T_I + T_O$.

Optimization in the DP-slave (T_I)
The acquisition and update of the value of the Input Data of all involved DP-slaves should be done at the end of each Isochronous DP cycle as close as possible to ensure to provide the most actual values of the Input Data.

Optimization in the DP-slave (T_O)
The output of the value of the Output Data of all involved DP-slaves should be done at the end of the cyclic part (DX) of each Isochronous DP cycle as close as possible to ensure to provide the actual values for the Output Data as soon as possible.

Optimization in the DP-master (Class 1) (T_M)
The processing of the application tasks in the DP-master (Class 1) may be shifted at the end of the cyclic part of the Isochronous DP cycle (T_{DX}) by the parameter T_M. At this time the values of the Input Data of all involved DP-slaves has been update in the DP-master (Class 1). The processing of the application tasks in the DP-master (Class 1) has to be finished within the actual Isochronous DP cycle to provide new values for the Output Data of all involved DP-slaves to be conveyed at the next Isochronous DP cycle.

6.2.2.4.3 Application model of the DP-master (class 1) with isochronous mode functionality

6.2.2.4.3.1 Model variants

Two different models concerning the interface between the DP Application Layer and the application of the DP-master (Class 1) itself are provided by fieldbus DP for the DP-master (Class 1) supporting the Isochronous Mode:

- Buffered Synchronized
- Enhanced Synchronised

The DP Application Layer in conjunction with the data-link layer provides the necessary events and messages to synchronies the application of the DP-master (Class 1) with the DP Application Layer for both synchronization mechanism.

6.2.2.4.3.2 Buffered synchronized isochronous mode

The Buffered Synchronized model provides intermediate buffer to store the Input Data from and the Output Data for the DP-slaves during one or more Isochronous DP cycle. During this period the application may access these buffers to read and update their contents at any time. This mechanism requires less processing performance of the application tasks and less synchronization between the application and the DP Application Layer.

At first the application shall read the value of the Input Data with the Get Input service and shall update the value of the Output Data of the DP-slaves with the Set Output service. To inform the DP Application Layer to change the intermediate buffer the application has to set

the parameter Final of the Set Output service to TRUE at the update of the last DP-slave. The DP Application Layer exchanges the intermediate buffer at the end of the current Isochronous DP cycle. The application shall wait for the next SYNCH indication to start the next application cycle. The application has to ensure the right sequence of interactions.

Figure 18 illustrates the Buffered Synchronized Isochronous Mode at the DP-master (Class 1).

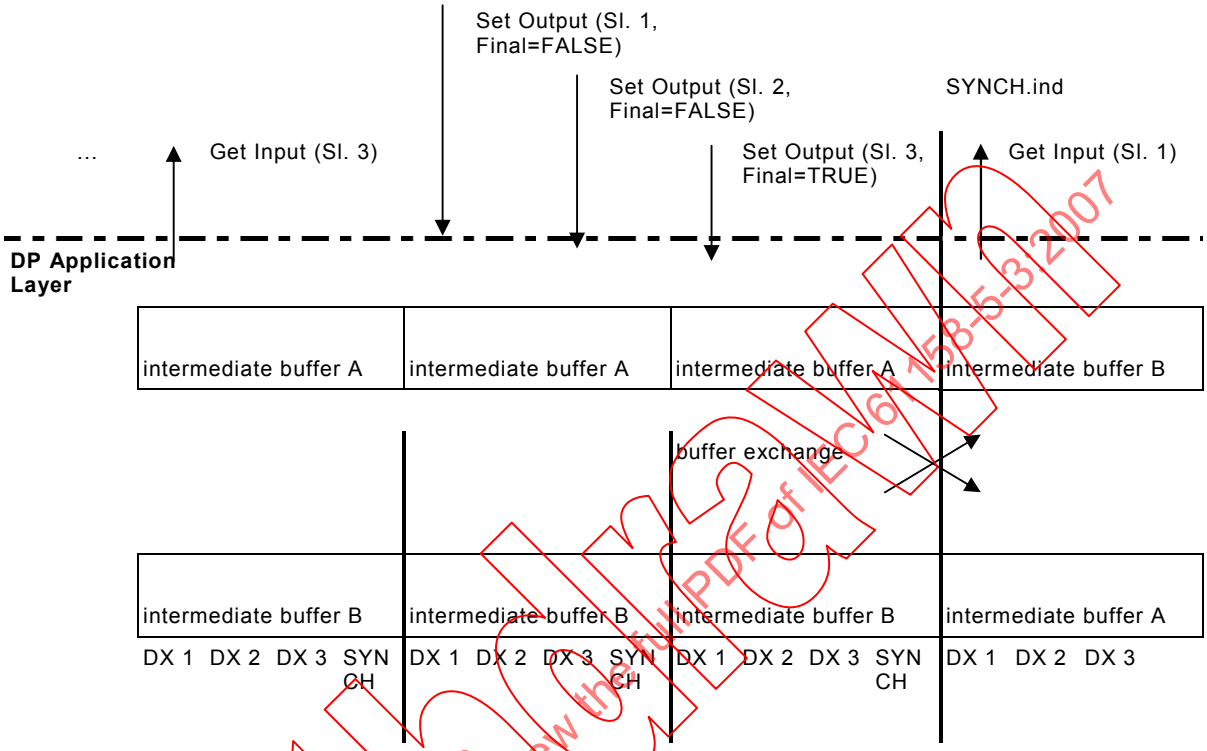

Figure 18 – Buffered synchronized isochronous mode at the DP-master (class 1)

6.2.2.4.3.3 Enhanced synchronized isochronous mode

The Enhanced Synchronized model is characterized by a well defined sequence of interactions between the application and the DP Application Layer of the DP-master (Class 1). The DP Application Layer provides means to supervise the proper sequence of interactions of the application within the Isochronous DP cycle and informs the application in case of erroneous interactions.

Figure 19 illustrates the Mechanism of the Enhanced Synchronized Isochronous Mode.

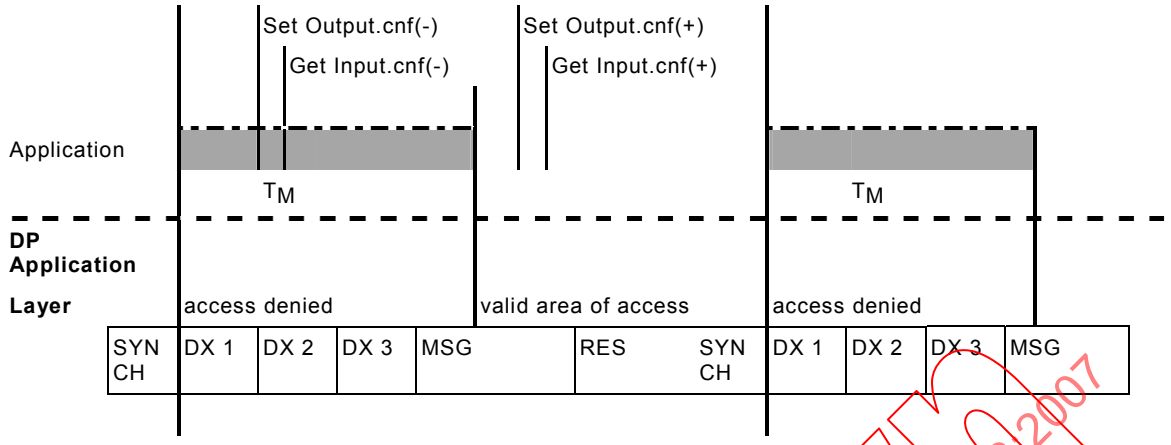

Figure 19 – Enhanced synchronized isochronous mode at the DP-master (class 1)

This mechanism requires enhanced processing performance of the application tasks and a very close synchronization between the application and the DP Application Layer.

6.2.2.4.4 Output data state machine description for isochronous mode

6.2.2.4.4.1 General

The behavior of a Output Data object is defined by the Output Data and PLL state machine in case of Isochronous Mode.

There are several State Machines for Input and Output Data objects. Figure 20 shows the interactions of PLL, Input and Output state machine.

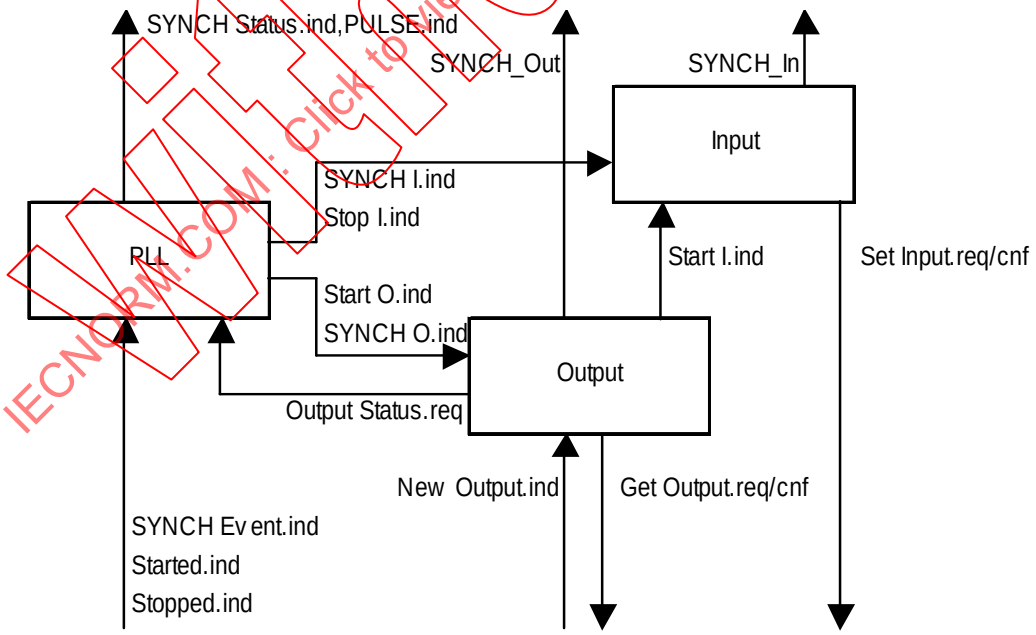

Figure 20 – Input, output and PLL state machine interaction

The Output Data and PLL state machine represents a part of the User functionality and are included to ensure interoperability.

The Output Data and PLL state machine provides events to other parts of the User functionality. The Output Data and PLL state machine have interfaces to the Input Data state machine for co-ordination purposes.

The main functionalities of the Output Data state machine are:

- Transfer the received Output Data to the application on time.
- Check the correct sequence: SYNCH, New Output, read out or the Output Data at To-To_min.

Check the Masters Sign of Life (MLS).

The main functionalities of the PLL state machine are:

- Jitter compensation of the SYNCH-Event.ind in the DP-slave
- Drift compensation

Generation of a Pulse Scheme for the local application

The synchronization primitive (SYNCH) is conveyed by a Global Control service sent as multicast with fixed settings of the parameter Group Select and Control Command Freeze and/or Sync.

The synchronization (SYNCH) indicated to the application of the DP-slave may jitter, may be lost at the conveyance to the DP-slave or may be received too late because the DP-master (Class 1) has sent the synchronization message not in time. A PLL (built in hardware or software) shall be used in the DP-slaves to minimize the jitters and phase shifts of the synchronization message and to compensate the loss of synchronization messages and may reduce the possible variation of the synchronization to a value the DP-slave operating in the Isochronous Mode can tolerate. If the variation exceeds a certain limit the application of the DP-slave will leave the Isochronous Mode an will change to a safe state.

The PLL in the DP-slave shall handle the following situations

- The jitters of the synchronization message at the DP-slave have to be smoothed by the PLL. If the jitters exceeds a certain limit, the PLL will recognize a lost clock.
- Synchronization messages lost due to bus disturbances (the multicast may not be retried) have to be compensated.
- Drift: After the start-up the Master will send Synchronization Messages. A DP-slave has to adjust the local time T_{DP} to the Masters T_{DP}. At first SYNCH the Window shall be set to first Window, which is at least T_{PLL_W}*n1+T_{DP}*n2 (n1>1, n2>0,0003). If the following SYNCH are within the Window, die local T_{DP} will be adjusted and the size of the Window shall be reduced until the window has the size of T_{PLL_W}. The PLL shall enter the normal Operating mode now. Additional Drifts due to temperature change or ageing will be adjusted in the normal operating mode. Drift changes or Jitters should result in a small change of the T_{DP}. To minimize changes in T_{DP} drift changes and Jitters shall be limited.

Phase shifts due to line delay between the different DP-slaves may be compensated, if the output of the PLL will be shifted in relation to the clock. Each DP-slave and the DP-master (Class 1) have to have an individual time shift to be synchronised with the DP-slave which have the longest line delay.

6.2.2.4.4.2 Primitive definitions

6.2.2.4.4.2.1 Primitives exchanged between the AL and the PLL state machine

Table 43 shows primitive the exchanged between the User and the PLL state machine.

Table 43 – Primitives issued by the AL to the PLL state machine

Primitive Name	Source	Associated parameters	Functions
SYNCH Event.ind	AL	AREP Status	
Started.ind	AL	AREP	
Stopped.ind	AL	AREP	

The parameters used with the primitives are described in the Service Specification of the Output Data object (see 6.2.2.3).

6.2.2.4.4.2.2 Primitives exchanged between the user and the PLL state machine

Table 44 show the primitive exchanged between the User and the PLL state machine.

Table 44 – Primitives issued by the user to the PLL state machine

Primitive Name	Source	Associated parameters	Functions
SYNCH Status.ind	PLL	Status	Reports change of Operation Mode, Warnings and Errors
PULSE.ind	PLL		Signal for Application Process to start a new cycle. This signal can be subdivided from TDP.(see Figure 16, TSAPC)

The parameter Status used with the primitive SYNCH Status.ind reports the different operation mode changes, warnings and errors. Table 45 shows the allowed values of Status.

Table 45 – Allowed values of Status

Value	Source	Meaning
PLL_STOP	PLL	PLL is stopped Device has to be resynchronized
PLL_SYNCH_ERR	PLL	PLL detects Synchronization Errors and has to be resynchronized
PLL_SYNCHED	PLL	PLL is synchronized with the Masters SYNCH
OUTPUT_SEQUENCE	Output	Output detects Sequence errors (Violation of the sequence: SYNCH > New Output > SYNCH_Out)
OUTPUT_RESETMLS	Output	Output MLS check detects MLS errors and resynchronizes MLS
OUTPUT_RUNMLS	Output	Output MLS check detects MLS in sequence

6.2.2.4.4.2.3 Primitives exchanged between the user and the input state machine

Table 46 show the primitive exchanged between the User and the Input state machine.

Table 46 – Primitives issued by the user to the input state machine

Primitive Name	Source	Associated parameters	Functions
SYNCH In	Input		Collects the Values of the Input User Data

6.2.2.4.4.2.4 Primitives exchanged between the user and the output state machine

Table 47 shows the primitive exchanged between the User and the Output state machine.

Table 47 – Primitives issued by the user to the output state machine

Primitive Name	Source	Associated parameters	Functions
SYNCH Out	Output		Pass the Output Buffer to the Output User Data

6.2.2.4.4.2.5 Primitives exchanged between the output state machine and the PLL state machine

Table 48 and Table 49 show the primitive exchanged between the Output state machine and the PLL state machine.

Table 48 – Primitives issued by the PLL to the output state machine

Primitive Name	Source	Associated parameters	Functions
Start_O.ind	PLL		Start Output Data processing
SYNCH_O.ind	PLL		Trigger Signal for Output Data Processing

Table 49 – Primitives issued by the output to the PLL state machine

Primitive Name	Source	Associated parameters	Functions
Output_Status.req	Output	Status	Report Status of Output Data processing

The parameter Status used with the primitive Output_Status.req reports the different warnings and errors.

6.2.2.4.4.2.6 Primitives exchanged between the input state machine and the PLL state machine

Table 50 shows the primitive exchanged between the Input state machine and the PLL state machine.

Table 50 – Primitives issued by the PLL to the input state machine

Primitive Name	Source	Associated parameters	Functions
Stop_I.ind	PLL		Stop Input Data processing
SYNCH_I.ind	PLL		Trigger Signal for Input Data Processing

6.2.2.4.4.2.7 Primitives exchanged between the input state machine and the output state machine

Table 51 shows the primitive exchanged between the Input state machine and the Output state machine.

Table 51 – Primitives issued by the output to the input state machine

Primitive Name	Source	Associated parameters	Functions
Start I.ind	Output		Start Output Data processing

6.2.2.4.4.2.8 Primitives exchanged between the AL and the output state machine

Table 52 shows the primitive exchanged between the Output state machine and AL.

Table 52 – Primitives issued by the output state machine to the AL

Primitive Name	Source	Associated parameters	Functions
Get Output.req	Output	AREP	

Table 53 shows the primitive exchanged between the AL and Output state machine.

Table 53 – Primitives issued by the AL to the output state machine

Primitive Name	Source	Associated parameters	Functions
Get Output.cnf	AL	AREP Output Data Clear Flag New Flag	
New Output.ind	AL	AREP Clear Flag	

6.2.2.4.4.2.9 Primitives exchanged between the input state machine and the AL

Table 54 shows the primitive exchanged between the Input state machine and the AL.

Table 54 – Primitives issued by the input state machine to the AL

Primitive Name	Source	Associated parameters	Functions
Set Input.req	Input	AREP Input Data	

Table 55 shows the primitive exchanged between the Input state machine and the AL.

Table 55 – Primitives issued by the AL to the input state machine

Primitive Name	Source	Associated parameters	Functions
Set Input.cnf	AL	AREP	

6.2.2.4.4.3 Local variable definitions of PLL

AREP

This local variable contains the AREP which represents the application relationships.

W

This local variable contains the current value of the window. The window is used to detect a valid SYNCH Event.ind.

Tdp_n

This local variable contains the current value of the adjusted cycle time. The adaptation is necessary due to clock drifts in a DP-master (class 1).

count_error

This local variable contains the current value of the weighted errors in relation to the SYNCH Event.ind within the window.

RUN

This local variable indicates that the Output State Machine is already started.

cnt

This local variable contains the current value of the segment counter

seg

This local variable contains the current value of the segment time (=Tdp_n/m)

TimerW

This local timer contains the current value of either the time between a SYNCH Event.ind and the beginning of the next window or the time between window start and SYNCH Event.ind. If there is no SYNCH Event.ind within the window, the end of the window substitutes this event. The value of the timer contains the time expired since the last start.

TimerP

This local timer contains the current value of either the time between time between two PULSE.ind. This is used to subdivide.

6.2.2.4.4.4 Local constant definitions of PLL

E_limit
This local constant limits the number of acceptable synchronization errors during a time interval.

N_error
This local constant is used to weight the synchronization errors and the synchronization success. A factor of 2 means that 2 SYNCH Event.ind are needed to compensate an synchronization error.

m
This local constant is used to subdivide Tdp_n.

6.2.2.4.4.5 Local function

NOTE Some of these functions are also used by the Output and Input state machine.

inc(x)
if (x=LIMIT) then x:=0 else x:=x+1

inc_LS(x)
if (x=LS_LIMIT) then x:=1 else x:=x+1

reduce_Window(W)
Reduces W as long as W>Tpllw. W shall not be smaller than Tpllw.

NOTE This function does not decrement W as linear function but can change in a discontinuous way (keeping W for a while and reducing it by dividing the value with 2)

adjust_Tdp(Tdp_n, error)
Adapt Tdp_n to a change of Tdp. To avoid unnecessary changes due to jitters an error should result in a change of Tdp_n smaller than the error value.

6.2.2.4.4.6 PLL state diagram

Figure 21 shows the state diagram of the PLL state machine.

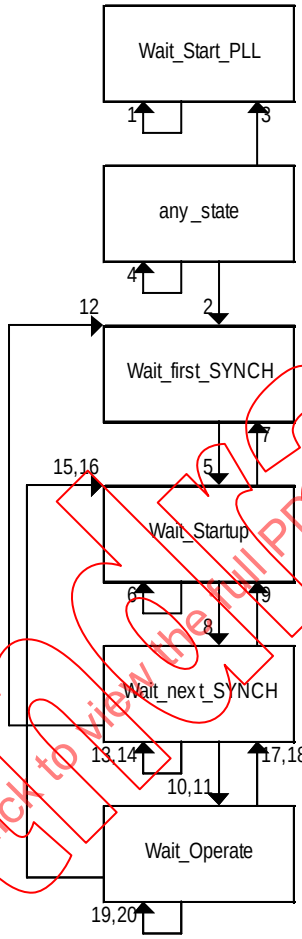

Figure 21 – PLL state diagram

6.2.2.4.4.7 PLL state table

Table 56 shows the states of a PLL state machine.

Table 56 – PLL state table

#	Current State	Event /Condition =>Action	Next State
1	Wait_Start_PLL	SYNCH Event.ind(AREP,Status) => ignore	Wait_Start_PLL
2	any_state	Started.ind(AREP) => W := first_Window Tdp_n := Tdp RUN := 0 count_error := 0	Wait_first_SYNCH
3	any_state	Stopped.ind(AREP) => Status := PLL_STOP SYNCH Status.ind(Status) Stop_I.ind	Wait_Start_PLL
4	any_state	Output Status.req(Status) => SYNCH Status.ind(Status) if (Status = OUTPUT_RESETMLS) Stop_I.ind endif	any_state
5	Wait_first_SYNCH	SYNCH Event.ind(AREP,Status) => start TimerW(Tdp_n-W)	Wait_Startup
6	Wait_Startup	SYNCH Event.ind(AREP,Status) /count_error < e_limit => count_error := count_error + 1	Wait_Startup
7	Wait_Startup	SYNCH Event.ind(AREP,Status) /count_error = e_limit => Status := PLL_SYNCH_ERR W := first_Window Tdp_n := Tdp count_error := 0 start TimerW(Tdp_n-W) SYNCH Status.ind(Status) Stop_I.ind	Wait_first_SYNCH
8	Wait_Startup	TimerW expired => start TimerW(2*W)	Wait_next_SYNCH
9	Wait_next_SYNCH	SYNCH Event.ind(AREP,Status) /W > Tpllw => Tdp_n:= adjust_Tdp(Tdp_n ,(TimerW-W)) W:=reduce_Window(W) start TimerW(Tdp_n-TimerW)	Wait_Startup
10	Wait_next_SYNCH	SYNCH Event.ind(AREP,Status) /W = Tpllw => Tdp_n:= adjust_Tdp(Tdp_n ,(TimerW-W)) count_error := 0start TimerW(Tdp_n-TimerW)	Wait_Operate
11	Wait_next_SYNCH	TimerW expired /count_error < e_limit => count_error := count_error + 1 start TimerW(Tdp-2*W)	Wait_Operate

#	Current State	Event /Condition =>Action	Next State
12	Wait_next_SYNCH	TimerW expired /count_error = e_limit => Status := PLL_SYNCH_ERR W := first_Window Tdp_n := Tdp RUN := 0 count_error := 0 start TimerW(Tdp_n-W) SYNCH Status.ind(Status) Stop_I.ind	Wait_first_SYNCH
13	Wait_next_SYNCH	TimerP expired /cnt < m => cnt:= cnt+1 Start TimerP(Seg) PULSE.ind	Wait_next_SYNCH
14	Wait_next_SYNCH	TimerP expired /cnt = m => PULSE.ind	Wait_next_SYNCH
15	Wait_Operate	SYNCH Event.ind(AREP,Status) /count_error < e_limit => count_error := count_error + 1	Wait_Startup
16	Wait_Operate	SYNCH Event.ind(AREP,Status) /count_error = e_limit => Status := PLL_SYNCH_ERR W := first_Window Tdp_n := Tdp RUN := 0 count_error := 0 start TimerW(Tdp_n-W) SYNCH Status.ind(Status) Stop_I.ind	Wait_Startup
17	Wait_Operate	TimerW expired /RUN = 1 => cnt:=1 Seg = (Tdp_n)/m Start TimerP(W+Delay) Start TimerW(2*W) SYNCH_I.ind SYNCH_O.ind	Wait_next_SYNCH
18	Wait_Operate	TimerW expired /RUN = 0 => Status := PLL_SYNCHED cnt:=1 Seg = (Tdp_n)/m Start TimerP(W+Delay) RUN:= 1 Start TimerW(2*W) Start_O.ind SYNCH_I.ind SYNCH_O.ind SYNCH_Status.ind(Status)	Wait_next_SYNCH
19	Wait_Operate	TimerP expired /cnt < m => cnt:= cnt+1 Start TimerP(Seg) PULSE.ind	Wait_Operate
20	Wait_Operate	TimerP expired /cnt = m => PULSE.ind	Wait_Operate

6.2.2.4.4.8 Local variable definitions of output

AREP

This local variable contains the AREP which represents the application relationships.

wrong_output_update

This local variable contains the current value of the weighted errors of output updates.

MLS_start_counter

This local variable contains the current value of correct MLS at start-up.

MLS_error_counter

This local variable contains the current value of the MLS errors.

TMAPC_counter

This local variable contains the current value of the number of cycles since the last MLS update.

old_MLS

This local variable contains the value of the last MLS.

MLS_State

This local variable contains the value sub-state for MLS calculation.

Possible values are:

- Wait_first_MLS Start-up of MLS
- Wait_next_MLS Wait for next following Start-up MLS

Wait_MLS Wait for MLS during normal operation

O_Buffer

This local variable contains the values of the Output Data received. It contains a optional variable MLS which contains the live sign. If this variable is not present the State "Check_MLS" has no Action.

TimerTo

This local timer contains the current value of the time between a SYNCH Event.ind and the transfer of the Output_Data to the Application Process.

Timer(To-Tomin)

This local timer contains the current value of the time between a SYNCH Event.ind and the transfer of the Output_Data to the Output State Machine.

6.2.2.4.4.9 Local constant definitions

limit

This local constant limits the number of acceptable output update errors during a time interval.

n

This local constant is used to weight wrong output updates and successful. A factor of 2 means that 2 successful updates are needed to compensate an erroneous update.

MLS_factor

This local constant is used define the number of cycles between increments of MLS.

n_TMAPC

This local constant is used to limit the errors of MLS.

n_MLS
This local constant is used for start-up to have a reasonable number of correct MLS.

6.2.2.4.4.10 Local function

inc(x)
if (x=LIMIT) then x:=0 else x:=x+1

6.2.2.4.4.11 OUTPUT state diagram

Figure 22 shows the state diagram of the OUTPUT state machine.

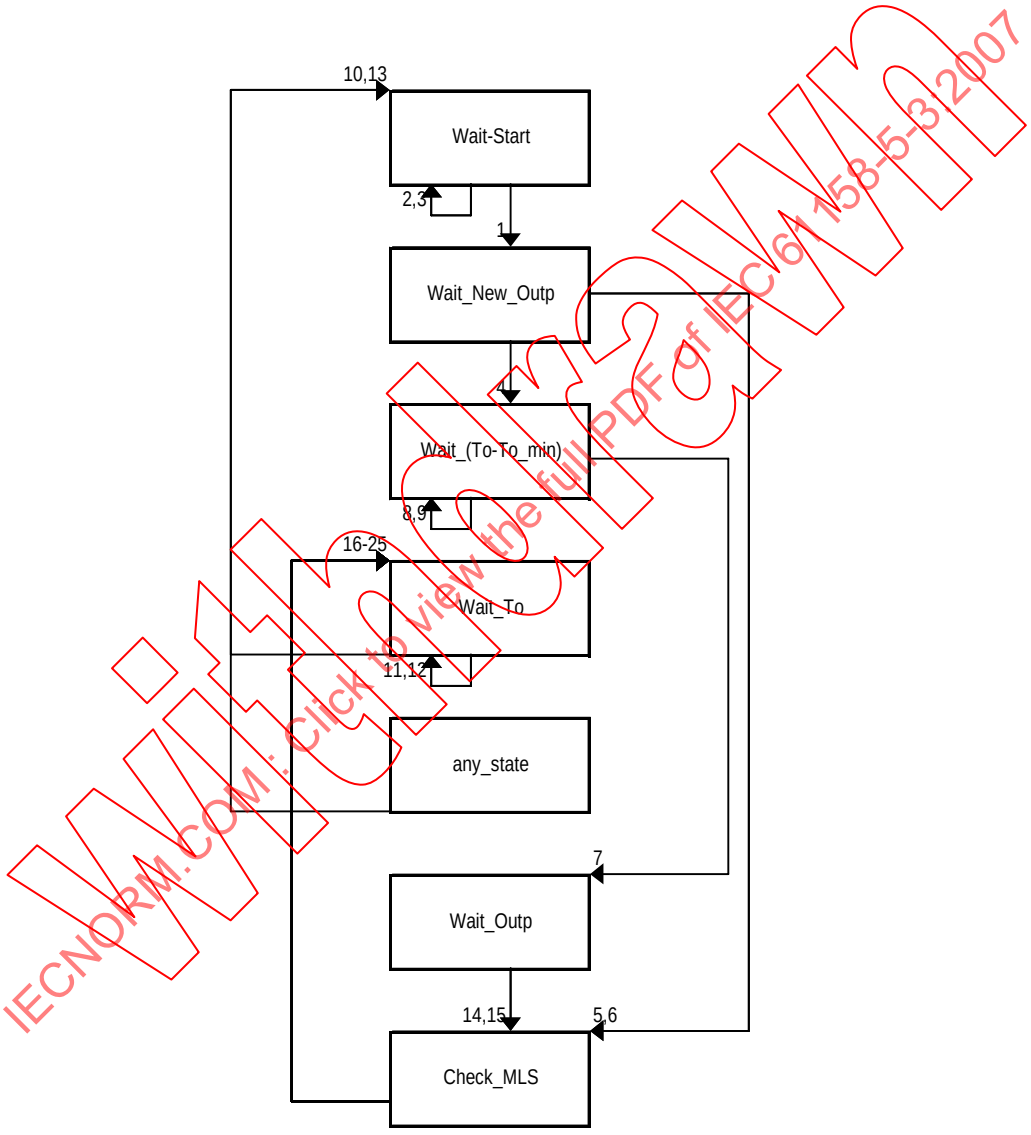

Figure 22 – OUTPUT state diagram

6.2.2.4.4.12 OUTPUT state table

Table 57 shows the states of a OUTPUT state machine.

Table 57 – OUTPUT state table

#	Current State	Event /Condition =>Action	Next State
1	Wait-Start	SYNCH_O.ind => Start Timer(To-To_min) Start TimerTo	Wait_New_Outp
2	Wait-Start	New_Output.ind (AREP, Clear_Flag) /wrong_output_upd < limit => wrong_output_upd= wrong_output_upd+n	Wait-Start
3	Wait-Start	New_Output.ind (AREP, Clear_Flag) /wrong_output_upd = limit => Status := OUTPUT_SEQUENCE Output_Status.req (Status)	Wait-Start
4	Wait_New_Outp	New_Output.ind (AREP, Clear_Flag) => if (wrong_output_upd>0) wrong_output_upd= wrong_output_upd-1	Wait_(To-To_min)
5	Wait_New_Outp	Timer(To-To_min) expired /wrong_output_upd < limit => wrong_output_upd= wrong_output_upd+n	Check_MLS
6	Wait_New_Outp	Timer(To-To_min) expired /wrong_output_upd = limit => Status := OUTPUT_SEQUENCE Output_Status.req (Status)	Check_MLS
7	Wait_(To-To_min)	Timer(To-To_min) expired => Get_Output.req(AREP)	Wait_Outp
8	Wait_(To-To_min)	New_Output.ind (AREP, Clear_Flag) /wrong_output_upd < limit => wrong_output_upd= wrong_output_upd+n	Wait_(To-To_min)
9	Wait_(To-To_min)	New_Output.ind (AREP, Clear_Flag) /wrong_output_upd = limit => Status := OUTPUT_SEQUENCE Output_Status.req (Status)	Wait_(To-To_min)
10	Wait_To	TimerTo expired => SYNCH_Out	Wait-Start
11	Wait_To	New_Output.ind (AREP, Clear_Flag) /wrong_output_upd < limit => wrong_output_upd= wrong_output_upd+n	Wait_To
12	Wait_To	New_Output.ind (AREP, Clear_Flag) /wrong_output_upd = limit => Status := OUTPUT_SEQUENCE Output_Status.req (Status)	Wait_To
13	any_state	Start_O.ind=>MLS_State:= Wait_first_MLSO_Buffer:=Nil(with O_Buffer.MLS:=0)old_MLS:=0MLS_error_counter:=0TMAPC_counter:=0MLS_start_counter:=0	Wait-Start
14	Wait_Outp	Get_Output.cnf (AREP, Output_Data, New_Flag, Clear_Flag) /New_Flag && !Clear_Flag => O_Buffer := Output_Data	Check_MLS
15	Wait_Outp	Get_Output.cnf (AREP, Output_Data, New_Flag, Clear_Flag) /!(New_Flag && !Clear_Flag) =>	Check_MLS

#	Current State	Event /Condition =>Action	Next State
16	Check_MLS	/MLS_State = Wait_first_MLS => MLS_State := Wait_next_MLS Status:=OUTPUT_RESETMLS old_MLS:=O_Buffer.MLS Output_Status.req (Status)	Wait_To
17	Check_MLS	/MLS_State = Wait_next_MLS TMAPC_counter < MLS_factor => MLS_State := Wait_next_MLS inc(TMAPC_counter)	Wait_To
18	Check_MLS	/MLS_State = Wait_next_MLS O_Buffer.MLS <> inc_LS(old_MLS) & TMAPC_counter = MLS_factor & MLS_error_counter < n_TMAPC => MLS_State := Wait_next_MLS inc_LS(old_MLS) TMAPC_counter:=0 MLS_error_counter := MLS_error_counter + n_error	Wait_To
19	Check_MLS	/MLS_State = Wait_next_MLS O_Buffer.MLS <> inc_LS(old_MLS) & TMAPC_counter = MLS_factor & MLS_error_counter >= n_TMAPC => MLS_State := Wait_first_MLS Status:=OUTPUT_RESETMLS TMAPC_counter:=0 MLS_error_counter:=0 old_MLS:=0 MLS_start_counter:=0 Output_Status.req (Status)	Wait_To
20	Check_MLS	/MLS_State = Wait_next_MLS O_Buffer.MLS = inc_LS(old_MLS) & TMAPC_counter = MLS_factor & MLS_start_counter < n_MLS => MLS_State := Wait_next_MLS TMAPC_counter:=0 if (MLS_error_counter >0) MLS_error_counter := MLS_error_counter –1 old_MLS:=MLS inc(MLS_start_counter)	Wait_To
21	Check_MLS	/MLS_State = Wait_next_MLS O_Buffer.MLS = inc_LS(old_MLS) & TMAPC_counter = MLS_factor & MLS_start_counter = n_MLS => MLS_State := Wait_MLS Status:=OUTPUT_RUNMLS TMAPC_counter:=0 MLS_error_counter:=0 old_MLS:=O_Buffer.MLS MLS_start_counter:=0 Output_Status.req (Status) Start_I.ind	Wait_To
22	Check_MLS	/MLS_State = Wait_MLS TMAPC_counter < MLS_factor => MLS_State := Wait_MLS inc(TMAPC_counter)	Wait_To

#	Current State	Event /Condition =>Action	Next State
23	Check_MLS	/MLS_State = Wait_MLS O_Buffer.MLS <> inc_LS(old_MLS) & TMAPC_counter = MLS_factor & MLS_error_counter < n_TMAPC => MLS_State := Wait_MLS inc_LS(old_MLS) TMAPC_counter:=0 MLS_error_counter := MLS_error_counter + n_error	Wait_To
24	Check_MLS	/MLS_State = Wait_MLS O_Buffer.MLS <> inc_LS(old_MLS) & TMAPC_counter = MLS_factor & MLS_error_counter >= n_TMAPC => MLS_State := Wait_first_MLS Status:=OUTPUT_RESETMLS TMAPC_counter:=0 MLS_error_counter:=0 old_MLS:=0 Output_Status.req (Status)	Wait_To
25	Check_MLS	/MLS_State = Wait_MLS O_Buffer.MLS = inc_LS(old_MLS) & TMAPC_counter = MLS_factor => MLS_State := Wait_MLS TMAPC_counter:=0 if (MLS_error_counter >0) MLS_error_counter := MLS_error_counter –1 old_MLS:=O_Buffer.MLS	Wait_To

6.2.2.4.5 Behavior of the input data object

For performance reasons, the value of the Input Data object should be transferred on change as fast as possible to the MS0 buffer for transmission. The speed and the jitters of this action is a performance parameter of a DP-slave. This parameter determines the freeze accuracy.

For highly synchronized DP-slaves, this action should be executed always at the same moment of time before the Freeze Command. This requires that a Freeze Command will be sent cyclically.

6.2.2.4.6 Input data state machine description for isochronous mode

6.2.2.4.6.1 Local variable definitions of input

AREP

This local variable contains the AREP which represents the application relationships.

SLS

This local variable contains the value of the Slave life sign.

I_Buffer

This local variable contains the values of the Input Data fetched from the Application Process. It contains a optional variable SLS which contains the live sign. If SLS is not present the Update of SLS is a NULL Operation.

TimerTix

This local timer contains the current value of the time between a SYNCH Event.ind and the transfer of the Input_Data to the AL.

Timer(TDP-Ti)

This local timer contains the current value of the time between a SYNCH Event.ind and the transfer of the Input_Data to the Input State Machine.

6.2.2.4.6.2 INPUT state diagram

Figure 23 shows the state diagram of the INPUT state machine.

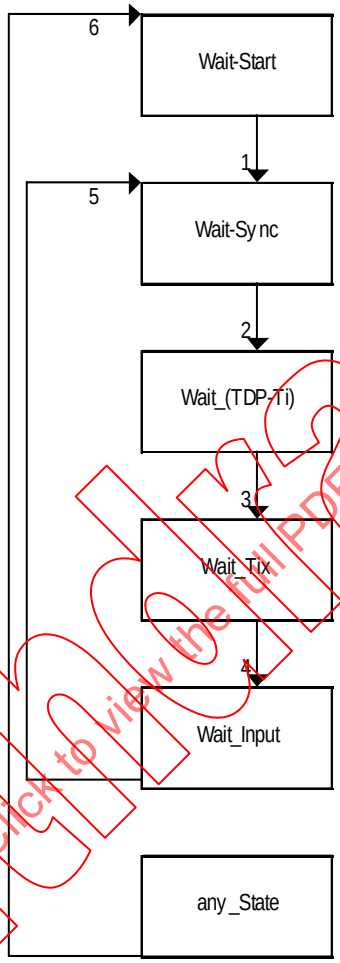

Figure 23 – INPUT state diagram

6.2.2.4.6.3 INPUT state table

Table 58 shows the states of a INPUT state machine.

Table 58 – INPUT state table

#	Current State	Event /Condition =>Action	Next State
1	Wait-Start	**Start_I.ind** =>	Wait-Sync
2	Wait-Sync	**SYNCH_I.ind** => Start Timer(TDP-Ti) Start TimerTix	Wait_(TDP-Ti)
3	Wait_(TDP-Ti)	**Timer(TDP-Ti) expired** => SYNCH_In	Wait_Tix
4	Wait_Tix	**TimerTix expired** => Input_Data:= I_Buffer Input_Data.SLS:= inc_LS(SLS) Set_Input.req (AREP,Input_Data)	Wait_Input
5	Wait_Input	**Set_Input.cnf(AREP)** =>	Wait-Sync
6	any_State	**Stop_I.ind** => SLS:= 0	Wait_Start

6.2.3 Diagnosis ASE

6.2.3.1 Overview

In the DP Application Layer environment, the Diagnosis ASE of a DP-slave contains Diagnosis objects that are used to report the status of the Default Application Process of the DP-slave and the modules which are part of the configuration. The Diagnosis ASE defines attributes of the Diagnosis objects and provides a set of local services used to set and get their values. The following Diagnosis objects are defined:

— Device Related Diagnosis object
— Status object
— Module Status Data object
— DXB-Link Status object
— Identifier Related Diagnosis object
— Channel Related Diagnosis object

The application of a DP-slave may provide the value of one or several Diagnosis objects to the MS0 AR. The DP-slave shall provide only the values of those Diagnosis objects whose values have changed since their last provision. The DP-slave may indicate (with the parameter Ext Diag Overflow) that the values of all Diagnosis objects with a changed value could not be provided with one Set Slave Diag service request. The DP-master (Class 1) shall indicate (with the parameter Ext Diag Overflow) in the Get Slave Diag confirmation that not all received values of the Diagnosis objects from the DP-slave could be provided to the application.

The DP-slave shall indicate (with the parameter Ext Diag Flag) that a relevant diagnosis event has occurred which leads to a change in at least one Diagnosis object (e. g. should be set if one element of the Identifier Related Diagnosis object is set to the value 1). This flag causes an entry in the attribute System Diagnosis of the Master Diag object of the DP-master (Class 1).

The DP-slave may indicate the actual version of the DP-slave with the parameter Revision Number provided by the Set Slave Diag service. The Revision Number may be used by a DP-master (Class 2) to find the related GSD file of the DP-slave.

The Diagnosis model uses the Client/Server access model. The Diagnosis model is based on a buffer to buffer relationship. Thus the Server application updates (sets) the values of the Diagnosis objects in its local buffer for the conveyance. The Client application may get the values of the Diagnosis objects from its local buffer.

The DP-master (Class 1) communicates via the MS0 AR with all Diagnosis objects of the assigned DP-slaves simultaneously. Additionally the Client application may communicate with the Diagnosis objects of a single DP-slave.

The DP-master (Class 2) communicates via the MS0 AR with the Diagnosis objects of a single DP-slave.

Furthermore additional Diagnosis information (attributes of the MS0 AR object) is provided by the DP Application Layer as parameters of the Get Slave Diag service and Read Slave Diag service.

The formal description of the various Diagnosis objects is presented next, followed by a description of its services.

6.2.3.2 Diagnosis class specification

6.2.3.2.1 Device related diagnosis class specification

6.2.3.2.1.1 Template

The Device Related Diagnosis object is described by the following template:

DP ASE: **Diagnosis ASE**
CLASS: **Device Related Diagnosis**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:
1. (m) Key Attribute: Implicit
2. (m) Attribute: Device Related Data Description

SERVICES:
1. (m) OpsService: Set Slave Diag
2. (m) OpsService: Get Slave Diag
3. (o) OpsService: Read Slave Diag
4. (o) OpsService: New Slave Diag

6.2.3.2.1.2 Attributes

Implicit
The attribute Implicit indicates that the Device Related Diagnosis object is implicitly addressed by the services.

Device Related Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used. The maximum length of this Device Related Data shall not exceed 62 octets.

6.2.3.2.1.3 Invocation of the device related diagnosis object

For the invocation of the Device Related Diagnosis object the following rules shall be considered:

- Device Related Diagnosis objects shall be used only if the device attribute DPV1 = FALSE.
- Device Related Diagnosis objects shall not exceed the total length of 62 octets. The attribute Device Related Data Description has to be set accordingly.

More than one Device Related Diagnosis object may be invoked in a DP-slave.

6.2.3.2.2 Identifier related diagnosis class specification

6.2.3.2.2.1 Template

This class specifies the state of the modules which are part of the configuration. Each bit is related to a Module (Slot). An element set to FALSE means that no diagnosis of the related module exists or no module is placed. A bit set to TRUE indicates a module failure. If a module failure exists, the values of the Input Data object and/or the Output Data object of this module are set to the default value by the DP-slave.

The Identifier Related Diagnosis object is described by the following template:

DP ASE: **Diagnosis ASE**
CLASS: **Identifier Related Diagnosis**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of Identifier Status
2.1.	(m)	Attribute:	Identifier Status

SERVICES:

1.	(m)	OpsService:	Set Slave Diag
2.	(m)	OpsService:	Get Slave Diag
3.	(o)	OpsService:	Read Slave Diag
4.	(o)	OpsService:	New Slave Diag

6.2.3.2.2.2 Attributes

Implicit
The attribute Implicit indicates that the Identifier Related Diagnosis object is implicitly addressed by the services.

List of Identifier Status
This attribute is composed of the following list elements:

Identifier Status
The elements of this list attribute indicates the state of each module. E.g. if the first module has an error the first element of the List of Identifier Status has the value 1.

The allowed values are shown in Table 59.

Table 59 – Identifier status

Value	Meaning
0	Module in operation.
1	Module not in operation.

6.2.3.2.2.3 Invocation of the identifier related diagnosis object

For the invocation of the Identifier Related Diagnosis object the following rules shall be considered:

- Only one Identifier Related Diagnosis object shall be invoked in a DP-slave.

According to the maximum number of modules, that are part of the configuration, an Identifier Related Diagnosis object shall not exceed the total length of 32 octets.

6.2.3.2.3 Channel related diagnosis class specification

6.2.3.2.3.1 Template

The Channel Related Diagnosis object is described by the following template:

DP ASE:			**Diagnosis ASE**
CLASS:		**Channel Related Diagnosis**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attributes:	Identifier
2.	(m)	Attribute:	Channel Type
3.	(m)	Attribute:	IO Type
4.	(m)	Attribute:	Error Type
SERVICES:			
1.	(m)	OpsService:	Set Slave Diag
2.	(m)	OpsService:	Get Slave Diag
3.	(o)	OpsService:	Read Slave Diag
4.	(o)	OpsService:	New Slave Diag

6.2.3.2.3.2 Attributes

Identifier
This key attribute is composed of Identifier Number and Channel Number to indicate the source of the Channel Related Diagnosis object.

Allowed values: 0 to 63, 0 to 63

Channel Type
This attribute indicates the channel-size to which the Channel Related Diagnosis object is related.

The allowed values are shown in Table 60.

Table 60 – Channel type

Value	Meaning
0	unspecific
1	1 bit
2	2 bit
3	4 bit
4	octet
5	word
6	2 words

IO Type
This attribute indicates the IO Type of the channel to which the Channel Related Diagnosis object is related.

The allowed values are shown in Table 61.

Table 61 – IO type

Value	Meaning
1	an Input channel
2	an Output channel
3	a Bi-directional channel

Error Type
This attribute indicates the error type of the Channel Related Diagnosis object.

The allowed values are shown in Table 62.

Table 62 – Error type

Value	Meaning
1	short circuit
2	undervoltage
3	overvoltage
4	overload
5	overtemperature
6	line break
7	upper limit value exceeded
8	lower limit value exceeded
9	error
16 to 31	manufacturer specific

6.2.3.2.3.3 Invocation of the channel related diagnosis object

For the invocation of the Channel Related Diagnosis object the following rule shall be considered:

Several Channel Related Diagnosis objects may be invoked in a DP-slave.

6.2.3.2.4 Status class specification

6.2.3.2.4.1 Template

The status model allows the transfer of status messages from the DP-slave to the DP-master (Class 1). The status model is slot based.

A Status object is described by the following template:

DP ASE: **Diagnosis ASE**
CLASS: **Status**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:
1. (m) Key Attributes: Identifier
2. (m) Attribute: Status Specifier
3. (o) Attribute: Status Data Description

SERVICES:
1. (m) OpsService: Set Slave Diag
2. (m) OpsService: Get Slave Diag
3. (o) OpsService: Read Slave Diag
4. (o) OpsService: New Slave Diag

6.2.3.2.4.2 Attributes

Identifier
This key attribute is composed of Slot Number and Status Type to indicate the number of the slot and the status type.

Allowed values for Slot Number: 0 to 254

The allowed values for Status Type are shown in Table 63.

Table 63 – Status type

Value	Meaning
1	Status Message
32-126	Manufacturer Specific

Status Specifier
This attribute specifies additional status information.

The allowed values are shown in Table 64.

Table 64 – Status specifier

Value	Meaning
0	no further differentiation
1	status appears
2	status disappears

Status Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

6.2.3.2.4.3 Invocation of the status object

For the invocation of the Status object the following rules apply:

- Status objects shall be used only if the device attribute DPV1 = TRUE
- Several Status objects may be invoked in a DP-slave.

A Status object shall not exceed the total length of 59 octets. The attribute Status Data Description has to be set accordingly.

6.2.3.2.5 Module status data class specification

6.2.3.2.5.1 Template

This class specifies the status of the modules which are part of the configuration.

The Module Status Data object is described by the following template:

DP ASE: **Diagnosis ASE**
CLASS: **Module Status Data**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Status Specifier
3.	(m)	Attribute:	List of Module Status

3.1.	(m)	Attribute:	Module Status
SERVICES:			
1.	(m)	OpsService:	Set Slave Diag
2.	(m)	OpsService:	Get Slave Diag
3.	(o)	OpsService:	Read Slave Diag
4.	(o)	OpsService:	New Slave Diag

6.2.3.2.5.2 Attributes

Implicit
The attribute Implicit indicates that the Module Status Data object is implicitly addressed by the services.

Status Specifier
This attribute specifies additional status information.

The allowed values are shown in Table 65.

Table 65 – Status specifier

Value	Meaning
0	no further differentiation
1	status appears
2	status disappears

List of Module Status
This attribute is composed of the following list elements:

Module Status
This attribute specifies additional status information for each module. The first element of the list contains the status of the first module (Slot Number 1).

The allowed values are shown in Table 66.

Table 66 – Module status

Value	Meaning
0	data valid
1	data invalid - signal error: the data of the corresponding module are not valid due to an error (inputs not accessible or outputs cannot be set, e.g. due to short circuit or overload)
2	data invalid - wrong module: the data of the corresponding module are not valid, due to a wrong module in place
3	data invalid - no module: the data of the corresponding module are not valid because there is no module in place

6.2.3.2.5.3 Invocation of the module status data object

For the invocation of the Module Status Data object the following rules apply:

- The Module Status Data object shall be used only if the device attribute DPV1 = TRUE.
- Only one Module Status Data object shall be invoked in a DP-slave.

The Module Status Data object shall not exceed the total length of 59 octets.

6.2.3.2.6 DXB-link status class specification

6.2.3.2.6.1 Template

This class specifies the status of the DXB-Links (Publisher/Subscriber relationships) contained in the DXB-Linktable at the Subscriber.

The DXB-Link Status object shall be used according the following rules:

- the DXB-Link Status object shall be issued at the first time with the Diagnosis information if the DP-slave has received a correct DXB-Linktable and if the DP-slave has entered the data transfer state
- the DXB-Link Status object shall be omitted if the DXB-Linktable was marked as invalid (e. g. receipt of new parameterisation data without a DXB-Linktable)

The DXB-Link Status object is described by the following template:

DP ASE: **Diagnosis ASE**
CLASS: **DXB-Link Status**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Status Specifier
3.	(m)	Attribute:	List of DXB-Link Status
3.1.	(m)	Attribute:	Publisher Address
3.2.	(m)	Attribute:	Link Status
3.3.	(m)	Attribute:	Link Error

SERVICES:

1.	(m)	OpsService:	Set Slave Diag
2.	(m)	OpsService:	Get Slave Diag
3.	(o)	OpsService:	Read Slave Diag
4.	(o)	OpsService:	New Slave Diag

6.2.3.2.6.2 Attributes

Implicit
The attribute Implicit indicates that the DXB-Link Status object is implicitly addressed by the services.

Status Specifier
This attribute specifies additional status information.

The allowed values are shown in Table 67.

Table 67 – Status specifier

Value	Meaning
0	no further differentiation
1	status appears
2	status disappears

List of DXB-Link Status
This attribute is composed of the following list elements:

Publisher Address
This attribute indicates the DL-address of the corresponding Publisher.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Link Status
This attribute indicates the status of the related DXB-Link.

The allowed values are shown in Table 68.

Table 68 – Link status

Value	Meaning
TRUE	DXB-Link active; at least one subscription of the related Publisher has been performed during the last watchdog control (T_{WD})
FALSE	DXB-Link not active; no subscription of the related Publisher has been performed during the last watchdog control (T_{WD}); the Subscriber may indicate detailed information about the reason with the attribute Link Error

Link Error
This attribute indicates detailed information about the failure of the related DXB-Link.

The allowed values are shown in Table 69.

Table 69 – Link error

Value	Meaning
0	unspecific
1	wrong length; the length of the Input Data of the Publisher is not in alignment with the value of the attribute Input Data Length Publisher of the related DXB-Link Entry

6.2.3.2.6.3 Invocation of the DXB-link status object

For the invocation of the DXB-Link Status object the following rules apply:

- The DXB-Link Status object shall be used only if the device attribute DPV1 is set to TRUE.
- Only one DXB-Link Status object shall be invoked in a DP-slave.

The DXB-Link Status object shall not exceed the total length of 59 octets. The attribute List of DXB-Link Status has to be set accordingly.

6.2.3.3 Diagnosis service specification

6.2.3.3.1 Set slave diag

This service shall be used by a DP-slave to pass the values of the Diagnosis objects to the DP Application Layer. The overall transmitted diagnosis information shall not exceed the total length of 238 octets. By this service the Server application process of the DP-slave updates the local buffer. According to the buffer to buffer characteristics of the MS0 AR the following behavior is possible:

- The Set Slave Diag requests come faster than the contents of the buffers are transported over the network. In this case not each value of the Diagnosis objects provided with the Set Slave Diag service is conveyed over the network. Only the latest values are transported, others are locally overwritten.
- The Set Slave Diag requests issued slower than the contents of the buffers are transported over the network. In this case the values of the Diagnosis objects provided with the Set Slave Diag service are conveyed more than once over the network.

If the Set Slave Diag requests are issued synchronous with the transport of the contents of the buffers each value of the Diagnosis objects provided with the Set Slave Diag service are conveyed over the network.

This service shall only be used in conjunction with the MS0 AR.

Table 70 shows the parameters of the service.

Table 70 – Set Slave Diag

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Ext Diag Overflow	M	
Ext Diag Flag	M	
Ext Diag Data	C	
Device Related Diagnosis Data	U	
List of Identifier Status	U	
Identifier Status	M	
List of Channel Related Diagnosis Data	U	
Identfier Number	M	
Channel Number	M	
Channel Type	M	
IO Type	M	
Error Type	M	
List of Status Data	U	
Slot Number	M	
Status Type	M	
Status Specifier	M	
Status Data	M	
Module Status Data	U	
Status Specifier	M	
List of Module Status	M	
Module Status	M	
DXB-Link Status	U	
Status Specifier	M	
List of DXB-Link Status	M	
Publisher Address	M	
Link Status	M	
Link Error	M	
Revision Number	U	
Result		S
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Ext Diag Flag
This parameter indicates that a relevant diagnosis event has occurred which leads to a change in at least one Diagnosis object (e. g. should be set if one element of the Identifier Related Diagnosis object is set to the value 1).

The allowed values are shown in Table 71.

Table 71 – Ext Diag Flag

Value	Meaning
TRUE	There exists at least one important diagnosis (such as short circuit). An entry has to be present in the Diagnosis objects (Device Related Diagnosis objects, Identifier Related Diagnosis object, Channel Related Diagnosis objects, Status objects, Module Status object).
FALSE	No important diagnosis entry exist. If the attribute value is FALSE, a status message (for example, supply voltage OK) may be present in the Diagnosis Data, or there exists no Diagnosis Data. The meaning of this status message is application specific.

Ext Diag Overflow
This Boolean parameter indicates that the values of all Diagnosis objects with a changed value could not been provided with this Set Slave Diag service request.

Allowed values: TRUE, FALSE

NOTE In case of an overflow condition it is allowed to truncate only at Diagnosis object limits.

Ext Diag Data
This parameter is composed of the following items

Device Related Diagnosis Data
This parameter contains the value of the Device Related Diagnosis object (this parameter shall be used only if DPV1 = FALSE).

List of Identifier Status
This parameter is composed of the following list elements:

Identifier Status
This parameter contains the value of the attribute Identifier Status of the Identifier Related Diagnosis object.

List of Channel Related Diagnosis Data
This parameter is composed of the following list elements:

Identifier Number
This parameter contains the value of the attribute Identifier Number of one Channel Related Diagnosis object.

Channel Number
This parameter contains the value of the attribute Channel Number of one Channel Related Diagnosis object.

Channel Type
This parameter contains the value of the attribute Channel Type of one Channel Related Diagnosis object.

IO Type
This parameter contains the value of the attribute IO Type of one Channel Related Diagnosis object.

Error Type
This parameter contains the value of one Channel Related Diagnosis object.

List of Status Data
This parameter is composed of the following list elements:

Slot Number
This parameter contains the value of the attribute Slot Number of one Status object.

Status Type
This parameter contains the value of the attribute Status Type for one Status object.

Status Specifier
This parameter contains the value of the attribute Status Specifier of one Status object.

Status Data
This parameter contains the value of one Status object.

Module Status Data
This parameter is composed of the following subparameters:

Status Specifier
This parameter contains the value of the attribute Status Specifier of the Module Status Data object.

List of Module Status
This parameter is composed of the following list elements:

Module Status
This parameter contains the value of the attribute Module Status of the Module Status Data object.

DXB-Link Status
This parameter is composed of the following subparameters:

Status Specifier
This parameter contains the value of the attribute Status Specifier of the DXB-Link Status object.

List of DXB-Link Status
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of the DXB-Link Status object.

Link Status
This parameter contains the value of the attribute Link Status of the DXB-Link Status object.

Link Error
This parameter contains the value of the attribute Link Error of the DXB-Link Status object.

Revision Number
This parameter contains the actual value of the Revision Number.

Result
This parameter indicates that the service request succeeded.

6.2.3.3.2 Get slave diag

This service may be used by a DP-master (Class 1) to read the previously received values of the Diagnosis objects of one or all assigned DP-slave which are related to the MS0 AR. Additionally the Get Slave Diag service provides the diagnosis information provided by the DP Application Layer of the DP-slave and the DP-master (Class 1). This service shall only be used in conjunction with the MS0 AR. Table 72 shows the parameters of the service.

Table 72 – Get Slave Diag

Parameter name	Req	Cnf
Argument	M	
AREP	M	
CREP	U	
Result(+)		S
AREP		M
CREP		C
List of Diag Data		M
Prm Req		M
Prm Fault		M
Cfg Fault		M
Not Supported		M
Station Not Ready		M
Stat Diag		M
WD On		M
Sync Mode		M
Freeze Mode		M
Master Add		M
Ident Number		M
Deactivated		M
Station Not Existent		M
Invalid Slave Response		M
Master Lock		M
Ext Diag Overflow		M
Ext Diag Flag		M
Ext Diag Data		C
Device Related Diagnosis Data		U
List of Identifier Status		U
Identifier Status		M
List of Channel Related Diagnosis Data		U
Identfier Number		M
Channel Number		M
Channel Type		M
IO Type		M
Error Type		M
List of Status Data		U
Slot Number		M

Parameter name	Req	Cnf
Status Type		M
Status Specifier		M
Status Data		M
Module Status Data		U
Status Specifier		M
List of Module Status		M
Module Status		M
DXB-Link Status		U
Status Specifier		M
List of DXB-Link Status		M
Publisher Address		M
Link Status		M
Link Error		M
Red Diagnosis		U
Red Status		S
Seq Number		M
Function		M
Primary Request		M
MS1 Command		M
Check Properties		M
Master State Clear		M
Red Status1		M
Role		M
HW Defect		M
Data Exchange		M
Safe State		M
DL Initialised		M
TOH started		M
Red Status2		M
Role		M
HW Defect		M
Data Exchange		M
Safe State		M
DL Initialised		M
TOH started		M
Prm Command Ack		S
Seq Number		M
Function		M
Primary Request		M
MS1 Command		M
Check Properties		M
Master State Clear		M
Red Status1		M
Role		M
HW Defect		M

Parameter name	Req	Cnf
Data Exchange		M
Master State Clear		M
DL Initialied		M
TOH started		M
Red Status2		M
Role		M
HW Defect		M
Data Exchange		M
Master State Clear		M
DL Initialised		M
TOH started		M
Revision Number		U
Result(-)		S
AREP		M
CREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
If this parameter is present then the Diagnosis objects of a single DP-slave are read.

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

List of Diag Data
This parameter is composed of the following list elements:

Prm Req
This parameter contains the value of the attribute Prm Req of the MS0 ARL entry of a DP-slave.

Prm Fault
This parameter contains the value of the attribute Prm Fault of the MS0 ARL entry of a DP-slave.

Cfg Fault
This parameter contains the value of the attribute Cfg Fault of the MS0 ARL entry of a DP-slave.

Not Supported
This parameter contains the value of the attribute Not Supported of the MS0 ARL entry of a DP-slave.

Station Not Ready
This parameter contains the value of the attribute Station Not Ready of the MS0 ARL entry of a DP-slave.

Stat Diag
This parameter contains the value of the attribute Stat Diag of the MS0 ARL entry of a DP-slave.

WD On
This parameter contains the value of the attribute WD On of the MS0 ARL entry of a DP-slave.

Sync Mode
This parameter contains the value of the attribute Sync Mode of the MS0 ARL entry of a DP-slave.

Freeze Mode
This parameter contains the value of the attribute Freeze Mode of the MS0 ARL entry of a DP-slave.

Master Add
This parameter contains the value of the attribute Master Add of the MS0 ARL entry of a DP-slave.

Ident Number
This parameter contains the value of the attribute Ident Number of the MS0 ARL entry of a DP-slave.

Deactivated
This parameter contains the value of the attribute Deactivated of the of the related CRL entry of the correspondent DP-slave of the DP-master (Class 1).

Station Non Existent
This parameter contains the value of the attribute Station Non Existent of the related CRL entry of the correspondent DP-slave of the DP-master (Class 1).

Invalid Slave Response
This parameter contains the value of the attribute Invalid Slave Response of the related CRL entry of the correspondent DP-slave of the DP-master (Class 1).

Master Lock
This parameter indicates whether the DP-slave is locked by a DP-master. If the Master Lock parameter is TRUE the DP-slave is locked by another DP-master. If the parameter is False the DP-slave is not locked by another DP-master (Class 1).

Ext Diag Overflow
This Boolean parameter contains the value of the parameter Ext Diag Overflow of the Set Slave Diag service if not all delivered values of the Diagnosis objects are contained in this service confirmation. This parameter contains the value of the attribute Ext Diag Overflow of the related CRL entry of the correspondent DP-slave of the DP-master (Class 1) if not all delivered values of the Diagnosis objects are contained in this service confirmation due to storage problems.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

NOTE In case of an overflow condition it is allowed to truncate only at Diagnosis object limits.

Ext Diag Flag
This parameter contains the value of the parameter Ext Diag Flag of the Set Slave Diag service of a DP-slave.

Ext Diag Data
This parameter is composed of the following subparameters:

Device Related Diagnosis Data
This parameter contains the value of the Device Related Diagnosis object of a DP-slave.

List of Identifier Status
This parameter is composed of the following list elements:

Identifier Status
This parameter contains the value of the attribute Identifier Status of the Identifier Related Diagnosis object of a DP-slave.

List of Channel Related Diagnosis Data
This parameter is composed of the following list elements:

Identifier Number
This parameter contains the value of the attribute Identifier Number of one Channel Related Diagnosis object of a DP-slave.

Channel Number
This parameter contains the value of the attribute Channel Number of one Channel Related Diagnosis object of a DP-slave.

Channel Type
This parameter contains the value of the attribute Channel Type of one Channel Related Diagnosis object of a DP-slave.

IO Type
This parameter contains the value of the attribute IO Type of one Channel Related Diagnosis object of a DP-slave.

Error Type
This parameter contains the value of the attribute Error Type of one Channel Related Diagnosis object of a DP-slave.

List of Status Data
This parameter is composed of the following list elements:

Slot Number
This parameter contains the value of the attribute Slot Number of one Status object of a DP-slave.

Status Type
This parameter contains the value of the attribute Status Type of one Status object of a DP-slave.

Status Specifier
This parameter contains the value of the attribute Status Specifier of one Status object of a DP-slave.

Status Data
This parameter contains the value of one Status object of a DP-slave.

Module Status Data
This parameter is composed of the following subparameters:

Status Specifier
This parameter contains the value of the attribute Status Specifier of the Module Status Data object of a DP-slave.

List of Module Status
This parameter is composed of the following list elements:

Module Status
This parameter contains the value of the attribute Module Status of the Module Status Data object of a DP-slave.

DXB-Link Status
This parameter is composed of the following subparameters:

Status Specifier
This parameter contains the value of the attribute Status Specifier of the DXB-Link Status object of a DP-slave.

List of DXB-Link Status
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of the DXB-Link Status object of a DP-slave.

Link Status
This parameter contains the value of the attribute Link Status of the DXB-Link Status object of a DP-slave.

Link Error
This parameter contains the value of the attribute Link Error of the DXB-Link Status object of a DP-slave.

Red Diagnosis
This parameter is composed of the following subparameters:

Red Status
This selection type parameter indicates that the parameter Red Diagnosis contains the diagnosis information of the redundant communication interfaces of a redundant DP-Slave. This parameter is composed of the following subparameters:

Seq Number
This parameter contains the Sequence Number necessary for the unique identification of the received Red Status.

Allowed values: 0 to 31

Function
This parameter is composed of the following subparameters:

Primary Request
This Boolean parameter set to TRUE indicates that a Primary Request was executed at this communication interface. This parameter set to FALSE indicates that no Primary Request was executed at this communication interface.

Allowed values: TRUE, FALSE

MS1 Command
This parameter indicates if a MS1 Command like Start MS1, Stop MS1, Reset MS1 or no MS1 Command was executed at this communication interface.

Allowed values: Start MS1, Stop MS1, Reset MS1, no Action

Check Properties
This Boolean parameter set to TRUE indicates that a Check Properties request was executed at this communication interface. This parameter set to FALSE indicates that no Check Properties request was executed at this communication interface.

Allowed values: TRUE, FALSE

Master State Clear
This Boolean parameter set to TRUE indicates that the related DP-Master (Class 1) have set the communication interface in the state

Clear. This parameter set to FALSE indicates that the related DP-Master (Class 1) have sent no Master State Clear command.

Allowed values: TRUE, FALSE

Red Status1
This parameter contains the redundancy status of the first communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that this communication interface acts as the primary interface. This parameter set to Backup indicates that this communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of this communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that this communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that this communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of this communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Red Status2
This parameter contains the redundancy status of the second (redundant) communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that the other communication interface acts as the primary interface. This parameter set to Backup indicates that the other communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of the other communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that the other communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that the other communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of the other communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Prm Command Ack
This selection type parameter indicates that the parameter Red Diagnosis contains the acknowledgment of a previously received Prm Command. This parameter is composed of the following subparameters:

Seq Number
This parameter contains the Sequence Number necessary for the unique identification of the received Prm Command Ack.

Allowed values: 0 to 31

Function
This parameter is composed of the following subparameters:

Primary Request
This Boolean parameter set to TRUE indicates that a Primary Request was executed at this communication interface. This parameter set to FALSE indicates that no Primary Request was executed at this communication interface.

Allowed values: TRUE, FALSE

MS1 Command
This parameter indicates if a MS1 Command like Start MS1, Stop MS1, Reset MS1 or no MS1 Command was executed at this communication interface.

Allowed values: Start MS1, Stop MS1, Reset MS1, no Action

Check Properties
This Boolean parameter set to TRUE indicates that a Check Properties request was executed at this communication interface. This parameter

set to FALSE indicates that no Check Properties request was executed at this communication interface.

Allowed values: TRUE, FALSE

Master State Clear
This Boolean parameter set to TRUE indicates that the related DP-Master (Class 1) have set the communication interface in the state Clear. This parameter set to FALSE indicates that the related DP-Master (Class 1) have sent no Master State Clear command.

Allowed values: TRUE, FALSE

Red Status1
This parameter contains the redundancy status of the first communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that this communication interface acts as the primary interface. This parameter set to Backup indicates that this communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of this communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that this communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that this communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of this communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Red Status2
This parameter contains the redundancy status of the second (redundant) communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that the other communication interface acts as the primary interface. This parameter set to Backup indicates that the other communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of the other communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that the other communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that the other communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of the other communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Revision Number
This parameter contains the actual value of the Revision Number.

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

6.2.3.3.3 Read slave diag

This service may be used by a DP-master (Class 2) to read acyclic the values of the Diagnosis objects of one DP-slave via a MS0 AR. Additionally the Read Slave Diag service provides the diagnosis information provided by the DP Application Layer of the DP-slave and the DP-master (Class 2).

Table 73 shows the parameters of the service.

Table 73 – Read Slave Diag

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result(+)		S
AREP		M
Diag Data		M
Prm Req		M
Prm Fault		M
Cfg Fault		M
Not Supported		M
Station Not Ready		M
Stat Diag		M
WD On		M
Sync Mode		M
Freeze Mode		M
Master Add		M
Ident Number		M
Ext Diag Overflow		M
Ext Diag Flag		M
Ext Diag Data		C
Device Related Diagnosis Data		U
List of Identifier Status		U
Identifier Status		M
List of Channel Related Diagnosis Data		U
Identfier Number		M
Channel Number		M
Channel Type		M
IO Type		M
Error Type		M
List of Status Data		U
Slot Number		M
Status Type		M
Status Specifier		M
Status Data		M
Module Status Data		U
Status Specifier		M
List of Module Status		M
Module Status		M
DXB-Link Status		U
Status Specifier		M
List of DXB-Link Status		M
Publisher Address		M
Link Status		M

Parameter name	Req	Cnf
Link Error		M
Red Diagnosis		U
Red Status		S
Seq Number		M
Function		M
Primary Request		M
MS1 Command		M
Check Properties		M
Master State Clear		M
Red Status1		M
Role		M
HW Defect		M
Data Exchange		M
Safe State		M
DL Initialised		M
TOH started		M
Red Status2		M
Role		M
HW Defect		M
Data Exchange		M
Safe State		M
DL Initialised		M
TOH started		M
Prm Command Ack		S
Seq Number		M
Function		M
Primary Request		M
MS1 Command		M
Check Properties		M
Master State Clear		M
Red Status1		M
Role		M
HW Defect		M
Data Exchange		M
Master State Clear		M
DL Initialied		M
TOH started		M
Red Status2		M
Role		M
HW Defect		M
Data Exchange		M
Master State Clear		M
DL Initialised		M
TOH started		M
Revision Number		U

Parameter name	Req	Cnf
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

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

Diag Data
This parameter is composed of the following list elements:

Prm Req
This parameter contains the value of the attribute Prm Req of the MS0 ARL entry of the selected DP-slave.

Prm Fault
This parameter contains the value of the attribute Prm Fault of the MS0 ARL entry of the selected DP-slave.

Cfg Fault
This parameter contains the value of the attribute Cfg Fault of the MS0 ARL entry of the selected DP-slave.

Not Supported
This parameter contains the value of the attribute Not Supported of the MS0 ARL entry of the selected DP-slave.

Station Not Ready
This parameter contains the value of the attribute Station Not Ready of the MS0 ARL entry of the selected DP-slave.

Stat Diag
This parameter contains the value of the attribute Stat Diag of the MS0 ARL entry of the selected DP-slave.

WD On
This parameter contains the value of the attribute WD On of the MS0 ARL entry of the selected DP-slave.

Sync Mode
This parameter contains the value of the attribute Sync Mode of the MS0 ARL entry of the selected DP-slave.

Freeze Mode
This parameter contains the value of the attribute Freeze Mode of the MS0 ARL entry of the selected DP-slave.

Master Add
This parameter contains the value of the attribute Master Add of the MS0 ARL entry of the selected DP-slave.

Ident Number
This parameter contains the value of the attribute Ident Number of the MS0 ARL entry of the selected DP-slave.

Ext Diag Overflow
This Boolean parameter contains the value of the parameter Ext Diag Overflow of the Set Slave Diag service if not all delivered values of the Diagnosis objects are contained in this service confirmation. This parameter will be set to TRUE by the AP of the DP-master (Class 2) if not all delivered values of the Diagnosis objects are contained in this service confirmation due to storage problems.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

NOTE In case of an overflow condition it is allowed to truncate only at Diagnosis object limits.

Ext Diag Flag
This parameter contains the value of the parameter Ext Diag Flag of the Set Slave Diag service of the selected DP-slave.

Ext Diag Data
This parameter is composed of the following subparameters:

Device Related Diagnosis Data
This parameter contains the value of the Device Related Diagnosis object of the selected DP-slave.

List of Identifier Status
This parameter is composed of the following list elements:

Identifier Status
This parameter contains the value of the attribute Identifier Status of the Identifier Related Diagnosis object of the selected DP-slave.

List of Channel Related Diagnosis Data
This parameter is composed of the following list elements:

Identifier Number
This parameter contains the value of the attribute Identifier Number of one Channel Related Diagnosis object of the selected DP-slave.

Channel Number
This parameter contains the value of the attribute Channel Number of one Channel Related Diagnosis object of the selected DP-slave.

Channel Type
This parameter contains the value of the attribute Channel Type of one Channel Related Diagnosis object of the selected DP-slave.

IO Type
This parameter contains the value of the attribute IO Type of one Channel Related Diagnosis object of the selected DP-slave.

Error Type
This parameter contains the value the attribute Error Type of one Channel Related Diagnosis object of the selected DP-slave.

List of Status Data
This parameter is composed of the following list elements:

Slot Number
This parameter contains the value of the attribute Slot Number of one Status object of the selected DP-slave.

Status Type
This parameter contains the value of the attribute Status Type of one Status object of the selected DP-slave.

Status Specifier
This parameter contains the value of the attribute Status Specifier of one Status object of the selected DP-slave.

Status Data
This parameter contains the value of one Status object of the selected DP-slave.

Module Status Data
This parameter is composed of the following subparameters:

Status Specifier
This parameter contains the value of the attribute Status Specifier of the Module Status Data object of the selected DP-slave.

List of Module Status
This parameter is composed of the following list elements:

Module Status
This parameter contains the value of the attribute Module Status of the Module Status Data object of the selected DP-slave.

DXB-Link Status
This parameter is composed of the following subparameters:

Status Specifier
This parameter contains the value of the attribute Status Specifier of the DXB-Link Status object of a DP-slave.

List of DXB-Link Status
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of the DXB-Link Status object of a DP-slave.

Link Status
This parameter contains the value of the attribute Link Status of the DXB-Link Status object of a DP-slave.

Link Error
This parameter contains the value of the attribute Link Error of the DXB-Link Status object of a DP-slave.

Red Diagnosis
This parameter is composed of the following subparameters:

Red Status
This selection type parameter indicates that the parameter Red Diagnosis contains the diagnosis information of the redundant communication interfaces of a redundant DP-Slave. This parameter is composed of the following subparameters:

Seq Number
This parameter contains the Sequence Number necessary for the unique identification of the received Red Status.

Allowed values: 0 to 31

Function
This parameter is composed of the following subparameters:

Primary Request
This Boolean parameter set to TRUE indicates that a Primary Request was executed at this communication interface. This parameter set to FALSE indicates that no Primary Request was executed at this communication interface.

Allowed values: TRUE, FALSE

MS1 Command
This parameter indicates if a MS1 Command like Start MS1, Stop MS1, Reset MS1 or no MS1 Command was executed at this communication interface.

Allowed values: Start MS1, Stop MS1, Reset MS1, no Action

Check Properties
This Boolean parameter set to TRUE indicates that a Check Properties request was executed at this communication interface. This parameter set to FALSE indicates that no Check Properties request was executed at this communication interface.

Allowed values: TRUE, FALSE

Master State Clear
This Boolean parameter set to TRUE indicates that the related DP-Master (Class 1) have set the communication interface in the state Clear. This parameter set to FALSE indicates that the related DP-Master (Class 1) have sent no Master State Clear command.

Allowed values: TRUE, FALSE

Red Status1
This parameter contains the redundancy status of the first communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that this communication interface acts as the primary interface. This parameter set to Backup indicates that this communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of this communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that this communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that this communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of this communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Red Status2
This parameter contains the redundancy status of the second (redundant) communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that the other communication interface acts as the primary interface. This parameter set to Backup indicates that the other communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of the other communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that the other communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that the other communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of the other communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Prm Command Ack
This selection type parameter indicates that the parameter Red Diagnosis contains the acknowledgment of a previously received Prm Command. This parameter is composed of the following subparameters:

Seq Number
This parameter contains the Sequence Number necessary for the unique identification of the received Prm Command Ack.

Allowed values: 0 to 31

Function
This parameter is composed of the following subparameters:

Primary Request
This Boolean parameter set to TRUE indicates that a Primary Request was executed at this communication interface. This parameter set to FALSE indicates that no Primary Request was executed at this communication interface.

Allowed values: TRUE, FALSE

MS1 Command
This parameter indicates if a MS1 Command like Start MS1, Stop MS1, Reset MS1 or no MS1 Command was executed at this communication interface.

Allowed values: Start MS1, Stop MS1, Reset MS1, no Action

Check Properties
This Boolean parameter set to TRUE indicates that a Check Properties request was executed at this communication interface. This parameter set to FALSE indicates that no Check Properties request was executed at this communication interface.

Allowed values: TRUE, FALSE

Master State Clear
This Boolean parameter set to TRUE indicates that the related DP-Master (Class 1) have set the communication interface in the state Clear. This parameter set to FALSE indicates that the related DP-Master (Class 1) have sent no Master State Clear command.

Allowed values: TRUE, FALSE

Red Status1
This parameter contains the redundancy status of the first communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that this communication interface acts as the primary interface. This parameter set to Backup indicates that this communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of this communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This Boolean parameter set to TRUE indicates that this communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that this communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of this communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Red Status2
This parameter contains the redundancy status of the second (redundant) communication interface of a redundant DP-Slave. This parameter is composed of the following subparameters:

Role
This parameter set to Primary indicates that the other communication interface acts as the primary interface. This parameter set to Backup indicates that the other communication interface acts as the Backup interface. This parameter set to Power On indicates that the assignment of the roles between the two communication interfaces is not finished yet (e. g. at Power On of the DP-Slave).

Allowed values: Primary, Backup, Power On

HW Defect
This Boolean parameter set to TRUE indicates that an HW Defect of the other communication interface has been detected.

Allowed values: TRUE, FALSE

Data Exchange
This vparameter set to TRUE indicates that the other communication interface is in the data exchange state.

Allowed values: TRUE, FALSE

Safe State
This Boolean parameter set to TRUE indicates that the other communication interface is in the Fail Safe state.

Allowed values: TRUE, FALSE

DL Initialised
This Boolean parameter set to TRUE indicates that the DL parameter of the other communication interface has been loaded.

Allowed values: TRUE, FALSE

TOH started
This Boolean parameter set to TRUE indicates that the Output Hold Timer of this communication interface has been started.

Allowed values: TRUE, FALSE

Revision Number
This parameter contains the value of the Revision Number of the selected DP-slave.

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

Status
This parameter contains the reason for the failure.

Allowed values: RE, DS, UE, RS, NA, NR, RR

6.2.3.3.4 New slave diag

This service may be used by a DP-master (Class 1) to indicate to the application that new received values of the Diagnosis objects of one assigned DP-slave which is related to the MS0 AR, is present. Additionally this service may be used to indicate that new diagnosis information provided by the DP Application Layer of the DP-slave or the DP-master (Class 1) is present.

Table 74 shows the parameters of the service.

Table 74 – New Slave Diag

Parameter name	Ind
Argument	M
AREP	M
CREP	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the related AR.

CREP
This parameter is the local identifier of the related CR (DP-slave).

6.2.4 Alarm ASE

6.2.4.1 Overview

The alarm model allows the transfer of an alarm from the DP-slave to the assigned DP-master (Class 1) and the explicit acknowledgement of the alarm by the DP-master (Class 1) via the MS1 AR.

Figure 24 shows the alarm interactions between the DP-slave (Alarm Notification) and the DP-master (Alarm Ack).

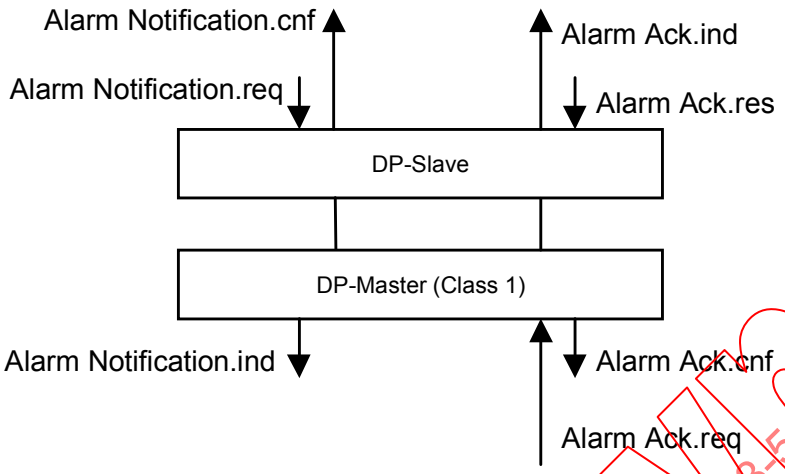

Figure 24 – Treatment of an alarm in the DP system

For the transmission of an alarm the following conditions have to be fulfilled:

- the DP-slave has to be in the data exchange mode
- the MS1 AR has to be open
- the corresponding alarm type has to be enabled

the limit of active alarms is not exceeded

In addition to the alarm transmission a Channel Related and/or Identifier Related Diagnosis may be issued which identify the exact source of the alarm.

Two alternatives of the alarm handling (see 6.2.10.3.1) are defined:

— only one alarm of a specific alarm type can be active at the same time (Type Mode)

— several alarms (2 to 32) independent of the type can be active at the same time (Sequence Mode)

The DP-master (Class 1) and the DP-slave shall implement the Type Mode and may additionally implement the Sequence Mode if alarms are supported. Alarms will be queued. The alarm remains active until the DP-master (Class 1) explicitly acknowledges the alarm.

The alarm acknowledgement is either performed via the CR, that conveys the Read and Write service, or optionally via a separate CR on the MS1 AR. With this latter feature, the alarms will not be delayed by read and write services.

The following alarm types are defined. They may be extended specific to the manufacturer:

Diagnosis Alarm
A diagnosis alarm signals an event within a module, for instance overtemperature, short circuit, etc.

Process Alarm
A process alarm signals the occurrence of an event in the connected process, for instance upper limit value exceeded.

Pull Alarm
A slot signals the withdrawal of a module.

Plug Alarm
A slot signals the insertion of a module.

Status Alarm
A status alarm signals a change in the state of a module, for instance run, stop or ready.

Update Alarm
An update alarm signals the change of a parameter in a module e.g. by a local operation or a remote access.

6.2.4.2 Alarm class specification

6.2.4.2.1 Template

An Alarm object is described by the following template:

DP ASE:			**Alarm ASE**
CLASS:		**ALARM DATA**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attributes:	Identifier
2.	(m)	Attribute:	Add Ack
3.	(m)	Attribute:	Alarm Specifier
4.	(o)	Attribute:	Alarm Data Description
SERVICES:			
1.	(m)	OpsService:	Alarm Notification
2.	(m)	OpsService:	Alarm Ack

6.2.4.2.2 Attributes

Identifier
This key attribute is composed of Slot Number and Alarm Type to indicate the number of the slot and the alarm type.

Allowed values for Slot Number: 0 to 254

The allowed values of Alarm Type are shown in Table 75.

Table 75 – Alarm type

Value	Meaning
1	Diagnosis Alarm
2	Process Alarm
3	Pull Alarm
4	Plug Alarm
5	Status Alarm
6	Update Alarm
32-126	Manufacturer Specific

Add Ack
This attribute indicates to the DP-master (Class 1) that this alarm requires an additional User acknowledgement. This can be done for instance by a write service. The alarm remains active after the Alarm Ack.

The allowed values are shown in Table 76.

Table 76 – Add Ack

Value	Meaning
TRUE	additional acknowledgement required
FALSE	no additional acknowledgement required

Alarm Specifier
This attribute specifies additional alarm information.

The allowed values are shown in Table 77.

Table 77 – Alarm specifier

Value	Meaning
0	no further differentiation
1	Alarm appears and the related module is disturbed
2	Alarm disappears and the related module has no further errors
3	Alarm disappears and the related module is still disturbed

Alarm Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

6.2.4.2.3 Invocation of the alarm data object

For the invocation of the Alarm Data object the following rules apply:

— Alarm Data objects shall be used only if the device attribute DPV1 = TRUE

— Several Alarm Data objects can be invoked in a DP-slave

— For each module only one Alarm Data object of one type shall be invoked

— Alarm Data objects shall not exceed the total length of 59 octets. The attribute Alarm Data Description has to be set accordingly.

6.2.4.3 Alarm service specification

6.2.4.3.1 Alarm notification

This service is used to transfer an alarm notification from the DP-slave to the DP-master (Class 1). This service shall only be used in conjunction with the MS1 AR. Table 78 shows the parameters of the service.

Table 78 – Alarm notification

Parameter name	Req	Ind	Cnf
Argument	M	M(=)	
AREP	M	M(=)	
Slot Number	M	M(=)	
Alarm Type	M	M(=)	
Seq Nr	M	M(=)	
Add Ack	M	M(=)	
Alarm Specifier	M	M(=)	
Alarm Data	U	C(=)	
Result(+)			S
AREP			M
Slot Number			M
Alarm Type			M
Seq Nr			M
Result(-)			S
AREP			M
Slot Number			M
Alarm Type			M
Seq Nr			M
Status			M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
This parameter contains the value of the attribute Slot Number of the Alarm Data object.

Alarm Type
This parameter contains the value of the attribute Alarm Type of the Alarm Data object.

Seq Nr
This parameter is used as an unique identifier for the transaction of an alarm notification by the DP Application Layer.

Allowed values: 0 to 31

Add Ack
This parameter contains the value of the attribute Add Ack of the Alarm Data object.

Alarm Specifier
This parameter contains the value of the attribute Alarm Specifier of the Alarm Data object.

Alarm Data
This parameter contains the value of the Alarm Data object.

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

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

Status
This parameter contains the reason for the failure.

Allowed values: MSAL1S_Stopped, Not_Enabled, Limit_Expired, Sequence_Nr_Pending, No_Start

6.2.4.3.2 Alarm ack

This service is used by the DP-master (Class 1) to acknowledge the reception of an alarm notification which has been previously received from the DP-slave. This service shall only be used in conjunction with the MS1 AR.

Table 79 shows the parameters of the service.

Table 79 – Alarm Ack

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
Alarm Type	M	M(=)		
Seq Nr	M	M(=)		
Result(+)			S	S
AREP			M	M(=)
Slot Number			M	M(=)
Alarm Type			M	M(=)
Seq Nr			M	M(=)
Result(-)			S	S
AREP			M	M(=)
Slot Number			M	M(=)
Alarm Type			M	M(=)
Seq Nr			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
This parameter contains the value of the attribute Slot Number of the of the previously received Alarm Data object.

Alarm Type
This parameter contains the value of the attribute Alarm Type of the previously received Alarm Data object.

Seq Nr
This parameter contains the sequence number of the previously received alarm notification.

Allowed values: 0 to 31

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

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

6.2.5 Context ASE

6.2.5.1 Overview

The Context ASE provides a set of services to parameterize and configure the Default AP of a DP-slave itself and its modules containing I/O data.

The Application Process of many DP-slaves need parameter (e.g. the range of an analogue signal) from the assigned DP-master (Class 1) to provide the data from the Input Data object and/or for the Output Data object according to the requirements of the AP of the DP-master (Class 1). These DP-slaves provide in the Context ASE a MS0 Parameter object or, for more flexibility, several MS0 Structured Parameter objects. The values for the attributes of these objects will be provided by the DP-master (Class 1) and assigned after validation from the AP of the DP-slave.

The Context ASE provides objects and the related services for the flexible parameterization of a DP-slave. The following objects are defined:

— MS0 User Parameter object (basic parameterization object)
— MS0 Structured User Parameter object
— DXB-Linktable object
— DXB-Subscribertable object
— IsoM Parameter object
— Time AR Parameter object

The functionality of a DP-slave application implies the necessary parameterization objects which have to be conveyed at the start-up phase of the DP system from the DP-master (Class 1) to the assigned DP-slaves via the MS0 AR. The application of the DP-slave has to check the parameterization data sent by the DP-master (Class 1). If the check fails the DP-slave may respond with an appropriate entry in the Diagnosis information.

The MS0 Configuration Elements object describes the module structure, the range of input and output data and contains information about the data consistency. The Context ASE defines a set of services used to set and get their values. The application of the DP-slave will compare the configuration data sent by the DP-master (Class 1) with the local configuration of the DP-slave. If they match the MS0 AR to this DP-slave will be established. The real configuration of the DP-slave may be read by a DP-master (Class 2) with the related service.

The DXB-Subscribertable object in conjunction with the MS0 Configuration Elements object provides the information the Subscriber needs to map the output data of the DP-master (Class 1) and/or the subscribed Input Data of one or more Publisher to the data areas of the Output Data Elements of the DP-slave.

The User shall check the contents of the DXB-Subscribertable provided with the Check User Prm service or Check Ext User Prm service and the configuration information provided with the Check Cfg service. In case of a positive result of these checks the User shall extract the necessary information from the DXB-Subscribertable the DP Application Layer needs for subscription. The User shall load this information with the Load CRL DXB Link Entries service.

For this mapping the following rules shall apply:

The data areas of an Output Data Element may be assigned:

— completely to the DP-master (Class 1) (default) or

— partially to the DP-master (Class 1) and partially to one or more Publisher or

— completely to one or more Publisher.

The lowest level of mapping information contained in the DXB-Subscribertable is one octet, no bit mapping will be supported.

It is allowed that for one or more data areas of an Output Data Element no values are provided from the DP-master (Class 1) or from a Publisher. For these data areas default values (zero) shall be used.

A data area of an Output Data Element shall always be supplied completely either by the DP-master (Class 1) or by a Publisher. It is not allowed to provide values only for a part of a data area of an Output Data Element.

Variations of the data areas (length of the Output Data) which are supplied by the DP-master (Class 1) may be achieved by variations of the Simple or Extended Output Data Description for this Output Data Element.

For those Output Data Elements of a DP-slave where no Output Data are required from the DP-master (Class 1) the attribute Length of the Simple or Extended Output Data Description (no Output Data) shall be set to zero. For these modules no Output Data will be transferred from the DP-master (Class 1).

The data areas of an Output Data Element which are usually assigned to the DP-master (Class 1) should be placed at the beginning of the Output Data Element. In this case no mapping information is necessary in the DXB-Subscribertable (default mapping).

The following checks may be accomplished for each module at the Subscriber depending on the information provided with the Check Cfg service and the information contained in the DXB-Subscribertable.

Case 1:

The length indicated in the Simple or Extended Output Description of a module is set to zero (no Output Data from the DP-master (Class 1)) and the DXB-Subscribertable contains at least one entry for this module then the following rules shall be applied:

— The sum of the length given in the attribute Length Data Area of the entries in the DXB-Subscribertable for the Simple or Extended Output Data Element shall be the length of the Simple or Extended Output Data Element, if there exist an entry for each data area of the Output Data Element.

— Overlapping of the values of the attributes Offset Data Area and Length Data Area of the entries in the DXB-Subscribertable for this Output Data Element is not allowed.

— The DXB-Subscribertable shall not contain any DXB Subscriber Entry of the DP-master (Class 1) (Master Data) for this Output Data Element.

— Not covered data areas of the Output Data Element shall be supplied with default values.

Case 2:

The length indicated in the Simple or Extended Output Description of a module is unequal to zero and smaller than the complete length of the Output Data Element (parts of the Output Data for this Output Data Element will be provided by the DP-master (Class 1)) and the DXB-Subscribertable contains at least one entry for this module then the following rules shall be applied:

— The sum of the values of the attributes Length Data Area of the DXB Subscriber Entries in the DXB-Subscribertable of the DP-master (Class 1) (Master Data) shall be equal the length given in the Simple or Extended Output Data Description of this Output Data Element.

— If the DXB-Subscribertable contains no entries for the DP-master (Class 1) (Master Data) the values for this Output Data Element from the DP-master (Class 1) shall be mapped to the first data areas of this Output Data Element (default mapping).

— The mapping of the subscribed data of the Publisher shall be done according to the entries in the DXB-Subscribertable.

— Overlapping of the values of the attributes Offset Data Area and Length Data Area of the entries in the DXB-Subscribertable for this Output Data Element is not allowed also not with the data areas supplied by the DP-master (Class 1) via the default mapping.

— Not covered data areas of the Output Data Element shall be supplied with default values.

Case 3:

The length indicated in the Simple or Extended Output Description of a module is equal to the length of the Output Data Element (all Output Data for this Output Data Element will be provided by the DP-master (Class 1)) then the following rule shall be applied:

— The DXB-Subscribertable shall contain no entries for this Output Data Element.

The Context ASE also provides the means to change the address of a DP-slave via the MS0 AR. This may be done by a DP-master (Class 2).

The length of an Input Data Element of a module shall not be influenced by the possible variations of assignments of the data areas of the Output Data Element.

The Context ASE is used to establish or release an association between an individual AP of a DP-master (Class 2) and an individual AP of the DP-slave for acyclic communication. This association exclusively belongs to the Process Data ASE.

The formal description of the Context ASE is presented next, followed by a description of its services.

6.2.5.2 Context class specification

6.2.5.2.1 MS0 user parameter class specification

6.2.5.2.1.1 Template

This class specifies the object for the parameter data that may be related to the DP-slave itself and/or to the modules of the DP-slave.

The MS0 User Parameter object is described by the following template:

DP ASE:			**Context ASE**
CLASS:		**MS0 User Parameter**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(o)	Attribute:	User Prm Data
2.1.	(o)	Attribute:	Device Related User Parameter
2.1.1	(m)	Attribute:	Data Description
2.2.	(o)	Attribute:	List of Module Related User Parameter
2.2.1	(m)	Attribute:	Data Description
SERVICES:			
1.	(m)	OpsService:	Check User Prm
2.	(m)	OpsService:	Check User Prm Result

6.2.5.2.1.2 Attributes

Implicit
The attribute Implicit indicates that the User Parameter object is implicitly addressed by the services.

User Prm Data
This attribute is an aggregate of the Device and/or List of Module Related User Parameter and is composed of the following elements.

Device Related User Parameter
This attribute is composed of following elements:

Data Description
This attribute contains either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

List of Module Related User Parameter
This attribute is composed of following list elements (each element corresponds to a module).

Data Description
This attribute contains either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

6.2.5.2.1.3 Invocation of the MS0 user parameter object

For the invocation of the MS0 User Parameter object the following rules apply:

— The MS0 User Parameter object shall not exceed the total length of 237 octets (only 234 if the device attribute DPV1 = TRUE). The attributes Data Description and List of Module Related User Parameter have to be set accordingly.

— The MS0 User Parameter object shall exist only if the device attribute DPV1 is set to TRUE and the parameter Prm Structure provided with the Check User Prm service is set to FALSE or if the device attribute DPV1 is set to FALSE.

— Only one MS0 User Parameter object can be invoked in a DP-slave.

6.2.5.2.2 MS0 structured user parameter class specification

6.2.5.2.2.1 Template

In contrast to the MS0 User Parameter object this class defines separate objects for the device itself and for each of it's modules. Furthermore these objects allow the unique identification and assignment.

This structuring should be used by enhanced devices with a high variation of parameters which may not be present in every application. This object is mandatory for DP-slaves which support the Isochronous Mode and for the conveyance of the DXB-Linktable.

The MS0 Structured User Parameter object is described by the following template:

DP ASE:			**Context ASE**
CLASS:		**MS0 Structured User Parameter**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Identifier
2.	(m)	Attribute:	Structured User Prm Data
2.1.	(s)	Attribute:	Device Related User Parameter
2.1.1	(m)	Attribute:	Data Description

2.2.	(s)	Attribute:	Module Related User Parameter
2.2.1	(m)	Attribute:	Data Description

SERVICES:

1.	(m)	OpsService:	Check User Prm
2.	(o)	OpsService:	Check Ext User Prm
3.	(m)	OpsService:	Check User Prm Result
4.	(o)	OpsService:	Check Ext User Prm Result

6.2.5.2.2.2 Attributes

Identifier
This key attribute is composed of Slot Number and Prm Data Type to indicate the number of the slot and the type of parameter data the specified MS0 Structured User Parameter object contains. If the attribute Slot Number contains the value 0 the following Structured Prm Data are Device Related User Parameter, otherwise they are Module Related User Parameter

Attribute Type of Slot Number: Unsigned8

Allowed values for Slot Number: 0 to 244

Attribute Type of Prm Data Type:

The allowed values of Prm Data Type are shown in Table 80.

Table 80 – Prm data type

Value	Meaning
129	Parameter Data
32–128	Manufacturer Specific

Structured User Prm Data
This attribute contains either Device or Module Related User Parameter indicated by the attribute Identifier and is composed of the following elements.

Device Related User Parameter
This attribute is composed of following elements:

Data Description
This attribute contains either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Module Related User Parameter
This attribute is composed of following elements.

Data Description
This attribute contains either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

6.2.5.2.2.3 Invocation of the MS0 structured user parameter object

For the invocation of the MS0 Structured User Parameter object the following rules apply:

— MS0 Structured User Parameter objects shall exist only if the device attribute DPV1 is set to TRUE.

— The MS0 Structured User Parameter objects may be provided with the Check User Prm service if the parameter Prm Structure provided with the Check User Prm service is set to TRUE.

— The description of all MS0 Structured User Parameter objects shall not exceed the total length of 230 octets if the values are provided with the Check User Prm service. The attribute Data Description has to be set accordingly.

— The description of all MS0 Structured User Parameter objects shall not exceed the total length of 240 octets if the values are provided with the Check Ext User Prm service. The attribute Data Description has to be set accordingly.

— Several MS0 Structured User Parameter objects may be invoked in a DP-slave.

— Only one MS0 Structured User Parameter object of the same type shall be invoked for each slot.

6.2.5.2.3 DXB-linktable class specification

6.2.5.2.3.1 Template

This class specifies the entries of the DXB-Linktable. Each entry specifies a Publisher the Subscriber has to subscribe from.

The DXB-Linktable object is described by the following template:

DP ASE:			**Context ASE**
CLASS:		**DXB-Linktable**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of DXB Link Entries
2.1.	(m)	Attribute:	Publisher Address
2.2.	(m)	Attribute:	Input Data Length Publisher
2.3.	(m)	Attribute:	Source Offset Publisher
2.4.	(m)	Attribute:	Length
SERVICES:			
1.	(m)	OpsService:	Check User Prm
2.	(o)	OpsService:	Check Ext User Prm
3.	(m)	OpsService:	Check User Prm Result
4.	(o)	OpsService:	Check Ext User Prm Result

6.2.5.2.3.2 Attributes

Implicit
The attribute Implicit indicates that the DXB-Linktable object is implicitly addressed by the services.

List of DXB Link Entries
This attribute is composed of the following list elements.

Publisher Address
This attribute indicates the DL-address of a DP-slave (Publisher) the Subscriber shall subscribe from. The following rules shall apply for this attribute:

A value between 0 and 125 represents a valid Publisher address and therefore an active DXB-Link.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Input Data Length Publisher
This attribute defines the number of octets of the Simple or Extended Input Data object of the related Publisher.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Source Offset Publisher
This attribute defines the beginning (offset) of the subscription within the Simple or Extended Input Data object of the related Publisher given by the attribute Publisher Address.

Attribute Type: Unsigned8

Allowed values: 0 to 243

Length
This attribute defines the number of octets the Subscriber shall subscribe from the related Publisher starting at the offset given in the attribute Source Offset Publisher.

Attribute Type: Unsigned8

Allowed values: 1 to 244

6.2.5.2.3.3 Invocation of the DXB-linktable object

For the invocation of the DXB-Linktable object the following rules apply:

— The DXB-Linktable object shall exist only if the device attribute DPV1 of the DP-slave is set to TRUE.

— Only one entry for each Publisher shall be contained in the DXB-Linktable.

— The values for the DXB-Linktable object may be provided with the Check User Prm service if the parameter Prm Structure provided with the Check User Prm service is set to TRUE.

— The description of the DXB-Linktable object shall not exceed the total length of 230 octets if the values of the DXB-Linktable object are provided with the Check User Prm service. The attribute List of DXB Link Entries has to be set accordingly.

— The description of the DXB-Linktable object shall not exceed the total length of 240 octets if the values of the DXB-Linktable object are provided with the Check Ext User Prm service. The attribute List of DXB Link Entries has to be set accordingly.

— Only one DXB-Linktable object shall be invoked in a DP-slave.

— The DXB-Linktable object shall only be invoked in a DP-slave if no DXB-Subscribertable object is invoked and vice versa.

6.2.5.2.4 DXB-subscribertable class specification

6.2.5.2.4.1 Template

This class specifies the entries of the DXB-Subscribertable. Each entry specifies the source, a Publisher or the DP-master (Class 1), and the related mapping information needed to transfer the correspondent data to the data areas of an Output Data Element of the related module.

The DXB-Subscribertable object is described by the following template:

DP ASE:			**Context ASE**
CLASS:		**DXB-Subscribertable**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of DXB Subscriber Entries
2.1.	(s)	Attribute:	Publisher Address
2.1.1.	(m)	Attribute:	Input Data Length Publisher
2.1.2.	(m)	Attribute:	Source Offset Publisher
2.2.	(s)	Attribute:	Master Data
2.2.1.	(m)	Attribute:	Source Offset Master
2.3.	(m)	Attribute:	Dest Slot Number
2.4.	(m)	Attribute:	Offset Data Area
2.5.	(m)	Attribute:	Length Data Area

SERVICES:

1.	(m)	OpsService:	Check User Prm
2.	(o)	OpsService:	Check Ext User Prm
3.	(m)	OpsService:	Check User Prm Result
4.	(o)	OpsService:	Check Ext User Prm Result

6.2.5.2.4.2 Attributes

Implicit
The attribute Implicit indicates that the DXB-Subscribertable object is implicitly addressed by the services.

List of DXB Subscriber Entries
This attribute is composed of the following list elements.

Publisher Address
This selection attribute indicates the DL-address of a DP-slave (Publisher) the Subscriber shall subscribe from. The following rules shall apply for this attribute:

A value between 0 and 125 represents a valid Publisher address and therefore an active DXB-Link.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Input Data Length Publisher
This attribute defines the number of octets of the Simple or Extended Input Data object of the related Publisher.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Source Offset Publisher
This attribute defines the beginning (offset) of the subscription within the Simple or Extended Input Data object of the related Publisher given by the attribute Publisher Address.

Attribute Type: Unsigned8

Allowed values: 0 to 243

Master Data
This selection attribute indicates that for this entry the data shall be derived from the related DP-master (Class 1).

Attribute Type: Unsigned8

Allowed values: 128

Source Offset Master
This attribute defines the beginning (offset) within the Output Data of the DP-master (Class 1).

Attribute Type: Unsigned8

Allowed values: 0 to 243

Dest Slot Number
This attribute indicates the Slot Number of the module to which the subscribed values of the Input Data object of the addressed Publisher or the values of the addressed Output Data of the related DP-master (Class 1) shall be mapped.

Attribute Type: Unsigned8

Allowed values: 0 to 244

Offset Data Area
This attribute defines the beginning (offset) within the Output Data of the module (data area) to which the subscribed values of the Input Data object of the addressed Publisher or the values of the addressed Output Data of the related DP-master (Class 1) shall be mapped.

Attribute Type: Unsigned8

Allowed values: 0 to 244

Length Data Area
This attribute defines the number of octets from the related Publisher the Subscriber shall subscribe to, starting at the offset given in the attribute Source Offset Publisher in case of a Publisher. If the data shall be derived from the DP-master (Class 1) the starting offset is given in the attribute Source Offset Master.

The subscribed values of the Input Data object of the addressed Publisher or the values of the addressed Output Data of the related DP-master (Class 1) shall be mapped to the data area of the module given in the attributes Dest Slot Number and Offset Data Area of the related DXB Subscriber Entry.

Attribute Type: Unsigned8

Allowed values: 1 to 244

6.2.5.2.4.3 Invocation of the DXB-subscribertable object

For the invocation of the DXB-Subscribertable object the following rules apply:

— The DXB-Subscribertable object shall exist only if the device attribute DPV1 of the DP-slave is set to TRUE.

— The values for the DXB-Subscribertable object may be provided with the Check User Prm service if the parameter Prm Structure provided with the Check User Prm service is set to TRUE.

— The description of the DXB-Subscribertable object shall not exceed the total length of 230 octets if the values of the DXB-Subscribertable object are provided with the Check User Prm service. The attribute List of DXB Link Entries has to be set accordingly.

— The description of the DXB-Subscribertable object shall not exceed the total length of 240 octets if the values of the DXB-Subscribertable object are provided with the Check Ext User Prm service. The attribute List of DXB Link Entries has to be set accordingly.

— Only one DXB-Subscribertable object shall be invoked in a DP-slave.

— The DXB-Subscribertable object shall only be invoked in a DP-slave if no DXB-Linktable object is invoked and vice versa.

6.2.5.2.5 IsoM parameter class specification

6.2.5.2.5.1 Template

This class specifies the necessary parameter for a DP-slave to operate in the Isochronous Mode.

The IsoM Parameter object is described by the following template:

DP ASE: **Context ASE**
CLASS: **IsoM Parameter**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:
1. (m) Key Attribute: Implicit

2.	(m)	Attribute:	T_{BASE_DP}
3.	(m)	Attribute:	T_{DP}
4.	(m)	Attribute:	T_{MAPC}
5.	(m)	Attribute:	T_{BASE_IO}
6.	(m)	Attribute:	T_I
7.	(m)	Attribute:	T_O
8.	(m)	Attribute:	T_{DX}
9.	(m)	Attribute:	T_{PLL_W}
10.	(m)	Attribute:	T_{PLL_D}

SERVICES:

1.	(m)	OpsService:	Check User Prm
2.	(o)	OpsService:	Check Ext User Prm
3.	(m)	OpsService:	Check User Prm Result
4.	(o)	OpsService:	Check Ext User Prm Result

6.2.5.2.5.2 Attributes

Implicit
The attribute Implicit indicates that the IsoM Parameter object is implicitly addressed by the services.

T_{BASE_DP}
This attribute indicates the time base (in units of 1/12µs) for the time T_{DP}.

Allowed values: 375, 750, 1 500, 3 000, 6 000, 12 000 (the value 1 500 is mandatory for standard Isochronous Mode)

T_{DP}
This attribute indicates the time T_{DP} (in units of T_{BASE_DP}. This time includes all parts of the Isochronous DP cycle:

— synchronization message (SYNCH)
— cyclic services (DX)
— acyclic services (MSG)
— token (TOK)
— spare time (RES)

Attribute Type: Unsigned16

Allowed values: 1 to $2^{16} - 1$

T_{MAPC}
This attribute indicates the Master Application Cycle Time (in units of T_{DP}) the application of the DP-master (Class 1) needs to process an application task completely. From Isochronous point of view there is exactly one user task to control the isochronous part of application.

Allowed values: 1 to 14

T_{BASE_IO}
This attribute indicates the time base (in units of 1/12µs) for the times T_I and T_O of the DP-slave.

Allowed values: 375, 750, 1 500, 3 000, 6 000, 12 000 (the value 1 500 is mandatory for standard Isochronous Mode)

T_I
This attribute indicates the time T_I (in units of T_{BASE_IO}) for the DP-slave.

Allowed values: 0 (special case), 1 to $2^{16} - 1$

T_O
This attribute indicates the time T_O (in units of T_{BASE_IO}) for the DP-slave.

Allowed values: 0 (special case), 1 to 2^{16} - 1

T_{DX}
This attribute indicates the time T_{DX} (in units of 1/12µs) for the DP-slave.

Allowed values: 0 (special case), 1 to 2^{32} - 1

T_{PLL_W}
This attribute indicates the time T_{PLL_W} (in units of 1/12µs) for the PLL of the DP-slave.

Allowed values: 1 to 2^{16} - 1 (default value 12)

T_{PLL_D}
This attribute indicates the time T_{PLL_D} (in units of 1/12µs) for the PLL of the DP-slave.

Attribute Type: Unsigned16

Allowed values: 0 to 2^{16} - 1

6.2.5.2.5.3 Invocation of the IsoM parameter object

For the invocation of the IsoM Parameter object the following rules apply:

- The IsoM Parameter object shall exist only if the device attribute DPV1 of the correspondent DP-slave is set to TRUE.
- The values of the IsoM Parameter object may be provided with the Check User Prm service if the parameter Prm Structure provided with the Check User Prm service is set to TRUE.
- The values of the IsoM Parameter object may be provided with the Check Ext User Prm service.

Only one IsoM Parameter object shall be invoked in a DP-slave.

6.2.5.2.6 Time AR parameter class specification

6.2.5.2.6.1 Template

This class specifies the necessary parameter for a DP-slave to operate as a Time Receiver.

The Time AR Parameter object is described by the following template:

DP ASE:			**Context ASE**
CLASS:		**Time AR Parameter**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Clock Sync Interval
3.	(o)	Attribute:	CS Delay Time
SERVICES:			
1.	(m)	OpsService:	Check User Prm
2.	(o)	OpsService:	Check Ext User Prm
3.	(m)	OpsService:	Check User Prm Result
4.	(o)	OpsService:	Check Ext User Prm Result

6.2.5.2.6.2 Attributes

Implicit
The attribute Implicit indicates that the Time AR Parameter object is implicitly addressed by the services.

Clock Sync Interval
This attribute contains the value for the attribute clock synchronization interval of the ARL of the DP-slave operating as time receiver.

Allowed values: 1 to 2^{16} - 1; Default value: 1 000

Time Base: 10 ms

CS Delay Time
This attribute contains the value for the attribute CS Delay Time of the ARL of the DP-slave operating as time receiver.

Default value: 0

Type: Network Time Difference

6.2.5.2.6.3 Invocation of the Time AR parameter object

For the invocation of the Time AR Parameter object the following rules have to be obeyed:

- The Time AR Parameter object shall exist only if the device attribute DPV1 of the correspondent DP-slave is set to TRUE.
- The values of the Time AR Parameter object may be provided with the Check User Prm service if the parameter Prm Structure provided with the Check User Prm service is set to TRUE.
- The values of the Time AR Parameter object may be provided with the Check Ext User Prm service.

Only one Time AR Parameter object shall be invoked in a DP-slave.

6.2.5.2.7 MS0 configuration elements class specification

6.2.5.2.7.1 Template

The MS0 Configuration Elements object is described by the following template:

DP ASE:			**Context ASE**
CLASS:			**MS0 Configuration Elements**
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of Cfg Data
2.1.	(o)	Attribute:	Input Data Reference
2.2.	(o)	Attribute:	Output Data Reference
2.3.	(o)	Attribute:	Manufacturer Specific Data
2.4.	(o)	Attribute:	Module Name
SERVICES:			
1.	(m)	OpsService:	Set Cfg
2.	(m)	OpsService:	Get Cfg
3.	(m)	OpsService:	Check Cfg
4.	(m)	OpsService:	Check Cfg Result

6.2.5.2.7.2 Attributes

Implicit
The attribute Implicit indicates that the MS0 Configuration Elements object is implicitly addressed by the services.

List of Cfg Data
This attribute is an aggregate of the configuration data of a DP-slave which is described by Cfg Identifier or Special Cfg Identifier. It shall contain at least one Cfg Data entry. A Cfg Data entry consists of the following attributes.

Input Data Reference
Reference to an Input Data Element of a Simple or an Extended Input Data object.

Output Data Reference
Reference to an Output Data Element of a Simple or an Extended Output Data object.

Manufacturer Specific Data
This attribute contains a manufacturer specific addition to the Input and/or Output Data Reference attributes. The contents are device specific or module specific.

Attribute Type: Octet string

Module Name
This attribute defines a name of a list element.

Attribute Type: Visible String(32)

6.2.5.2.7.3 Invocation of the MS0 configuration elements object

For the invocation of the MS0 Configuration Elements object the following rules apply:

- Only one MS0 Configuration Elements object shall be invoked in a DP-slave

The description of the MS0 Configurations Elements object shall not exceed 244 octets. The attribute List of Cfg Data has to be set accordingly.

6.2.5.2.8 Remanent parameter class specification

6.2.5.2.8.1 Template

A Remanent Parameter object is described by the following template:

DP ASE:			**Context ASE**
CLASS:			**Remanent Parameter**
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	No Add Change
3.	(m)	Attribute:	New Slave Add
4.	(o)	Attribute:	Rem Para
4.1	(m)	Attribute:	Data Description
SERVICES:			
1.	(m)	OpsService:	Set Slave Add

6.2.5.2.8.2 Attributes

Implicit
The attribute Implicit indicates that an Remanent Parameter object is implicitly addressed by the service.

No Add Change
This attribute specifies whether it is allowed to change the DP-slave address again. After the initial reset the DP-slave DL-address will be set to the default address (126) and the attribute No Add Change will be set to FALSE. To change this attribute from TRUE to FALSE an initial reset of the DP-slave is necessary.

Allowed values: FALSE, TRUE

New Slave Add
This attribute contains the address of the DP-slave which is to be set.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Rem Para
This attribute contains parameters to be stored permanently.

Data Description
This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

6.2.5.2.8.3 Invocation of Rem Para object

For the invocation of the Remanent Parameter object the following rules apply:

- Only one Remanent Parameter object shall be invoked in a DP-slave.

The Remanent Parameter object shall not exceed 242 octets. The attribute Data Description has to be set accordingly.

6.2.5.2.9 MS2 user parameter class specification

6.2.5.2.9.1 Template

The MS2 User Parameter object is described by the following template:

DP ASE: **Context ASE**
CLASS: **MS2 User Parameter**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Features Supported
2.1.	(m)	Attribute:	Supported Feature
3.	(m)	Attribute:	Profile Features Supported
3.1.	(m)	Attribute:	Supported Profile Feature
4.	(m)	Attribute:	Profile Ident Number

SERVICES:

1.	(m)	OpsService:	Initiate
2.	(m)	OpsService:	Abort

6.2.5.2.9.2 Attributes

Implicit
The attribute Implicit indicates that the MS2 User Parameter object is implicitly addressed by the services.

Features Supported
This attribute is composed of the following list elements:

Supported Feature
This attribute reflects the actual features on the MS2 AR of a DP-slave. The DP-master (Class 2) and the DP-slave inform each other about the supported service functionality. The DP-slave and the DP-master (Class 2) have the possibility to adjust their functionality according to the requirements. The User application of the DP-master has to decide if the reported functionality is sufficient or not.

The allowed values are shown in Table 81.

Table 81 – Supported feature

Value	Meaning
1	Read and Write service supported
2 to 16	reserved for further features

Profile Features Supported
This attribute is composed of the following list elements:

Supported Profile Feature
This attribute reflects the actual profile features on the MS2 AR of a DP-slave. The DP-master (Class 2) and the DP-slave inform each other about the supported profile features. Regarding the used profile definition, the profile is identified by the Profile Ident Number. The User application of the DP-master has to decide if the reported functionality is sufficient or not.

The allowed values are shown in Table 82.

Table 82 – Supported profile feature

Value	Meaning
1 to 16	feature to be defined by profiles

Profile Ident Number
The attribute Profile Ident Number contains the unique identifier of a profile definition. All devices using the same profile definition have to use the same Profile Ident Number. The Profile Ident Number will be taken from the pool of Ident Numbers for vendor specific or authorized profiles. The value 0 indicates that no profile is supported.

Attribute type: Unsigned16

6.2.5.2.9.3 Invocation of the MS2 user parameter object

For the invocation of the MS2 user parameter object the following rules have to be obeyed:

Only one MS2 user parameter object shall be invoked in a DP-slave.

6.2.5.2.10 Red Context class specification

6.2.5.2.10.1 Template

This class specifies the status of the redundant communication interfaces of a DP-slave and the state of a previous provided switchover command.

The following template describes the Red Context object:

DP ASE: **Context ASE**
CLASS: **Red Context**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:
1. (m) Key Attribute: Communication Interface Identifier
2. (m) Attribute: Role
SERVICES:
1. (m) OpsService: Check User Prm
2. (m) OpsService: Check User Prm Result

6.2.5.2.10.2 Attributes

Communication Interface Identifier

This attribute identifies uniquely the communication interface.

Role

This attribute defines the actual role of the communication interface.

Allowed values: Primary, Backup, Power On

Table 83 – Role

Value	Meaning
Primary	At the primary interface there exists no constraints between the communication interface and the ASE's.
Backup	In case of Backup the following items between the communication interface and the ASE's are active: Diagnosis ASE (all services) Context ASE (Check User Prm, Check Ext User Prm, Check Cfg) Process Data ASE (all services, only via MS2 AR) Load Region ASE (all services, only via MS2 AR) Function Invocation ASE (all services, only via MS2 AR)
Power On	In case of Power On the following items between the communication interface and the ASE's are active: Diagnosis ASE (all services) Context ASE (Check User Prm, Check Ext User Prm)

6.2.5.2.10.3 Invocation of the Red Context object

For the invocation of the Red Context object the following rules have to be obeyed.

— The Red Context object shall be used only if the device attribute DPV1 is set to TRUE.

— For each communication interface only one Red Context object shall be invoked.

6.2.5.3 Context service specification

6.2.5.3.1 Check user Prm

The DP Application Layer of the DP-slave notifies the application when a valid parameterization is received from the DP-master because this service requests a validation of the User Prm Data or List of Structured Prm Data. A further Check User Prm indication is only issued after the Check User Prm service is completed with a Check User Prm Result service.

The Check User Prm service shall be acknowledged negatively (the parameter Prm OK of the service Check User Prm Result shall be set to FALSE) by the application of the DP-slave if the following conditions are met:

— the parameter Prm Structure is set to FALSE and the parameterization data (User Prm Data) are not allowed or not acceptable by the application of the DP-slave

- the parameter Prm Structure is set to TRUE and at least one of the MS0 Structured User Parameter object, the DXB-Linktable object, the DXB-Subscribertable object, or the IsoM Parameter object does not exist

the parameter Prm Structure is set to TRUE and at least one value for a MS0 Structured User Parameter object, or for the DXB-Linktable object, or for the DXB-Subscribertable object, or for the IsoM Parameter object is not allowed or not acceptable by the application of the DP-slave.

Table 84 shows the parameters of the service.

Table 84 – Check user Prm

Parameter name	Ind
Argument	M
AREP	M
Prm Structure	M
User Prm Data	C
Device Prm Data	U
Module Prm Data	U
List of Structured Prm Data	C
Structured User Prm Data	U
Device Related User Parameter	U
Device Prm Data	M
Module Related User Parameter	U
Slot Number	M
Module Prm Data	M
DXB-Linktable	U
List of DXB Link Entries	M
Publisher Address	M
Input Data Length Publisher	M
Source Offset Publisher	M
Length	M
DXB-Subscribertable	U
List of DXB Subscriber Entries	M
Publisher Address	C
Input Data Length Publisher	M
Source Offset Publisher	M
Master Data	C
Source Offset Master	M
Dest Slot Number	M
Offset Data Area	M
Length Data Area	M
IsoM Parameter	U
T_{BASE_DP}	M
T_{DP}	M
T_{MAPC}	M
T_{BASE_IO}	M
T_I	M
T_O	M
T_{DX}	M
T_{PLL_W}	M
T_{PLL_D}	M
Prm Command	U
Seq Number	M
Primary Request	M
MS1 Command	M
Master State Clear	M
Check Properties	M
Primary Request used	M
MS1 Command used	M
Address Change	M
Address Offset	M
Output Hold Time	M
Time AR Parameter	U
Clock Sync Interval	M
CS Delay Time	U

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the AR.

Prm Structure
This parameter indicates which type of parameterization objects are provided by the service Check User Prm. This parameter shall only be evaluated if the device attribute DPV1 is set to TRUE.

The allowed values are shown in Table 85.

Table 85 – Prm structure

Value	Meaning
TRUE	The parameter which follow are related to the following objects: MS0 Structured User Parameter DXB-Linktable IsoM Parameter
FALSE	The parameter which follow are only related to the MS0 User Parameter object.

User Prm Data
This conditional parameter is only present if the parameter Prm Structure contains the value FALSE. It is composed of one or both of the following subparameters.

Device Prm Data
This parameter contains the value for the Device Related User Parameter of the MS0 User Parameter object.

Module Prm Data
This parameter contains the value for the Module Related User Parameter of the MS0 User Parameter object.

List of Structured Prm Data
This conditional parameter is only present if the parameter Prm Structure contains the value TRUE. This parameter is composed of one or more of the following list elements.

Structured User Prm Data
This parameter is composed of one or both of the following subparameters.

Device Related User Parameter
This parameter indicates the presence of Device Related User Parameter.

Device Prm Data
This parameter contains the value for the Device Related User Parameter of the MS0 Structured User Parameter object.

Module Related User Parameter
This parameter indicates the presence of Module Related User Parameter.

Slot Number
This parameter contains the value of the attribute Slot Number of the MS0 Structured User Parameter object.

Module Prm Data
This parameter contains the value for the Module Related User Parameter of the MS0 Structured User Parameter object.

DXB-Linktable
This parameter indicates the presence of the DXB-Linktable.

List of DXB Link Entries
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of one DXB-Linktable entry of the DXB-Linktable object.

Input Data Length Publisher
This parameter contains the value of the attribute Input Data Length Publisher of one DXB-Linktable entry of the DXB-Linktable object.

Source Offset Publisher
This parameter contains the value of the attribute Source Offset Publisher of one DXB-Linktable entry of the DXB-Linktable object.

Length
This parameter contains the value of the attribute Length of one DXB-Linktable entry of the DXB-Linktable object.

DXB-Subscribertable
This parameter indicates the presence of the DXB-Subscribertable.

List of DXB Subscriber Entries
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Input Data Length Publisher
This parameter contains the value of the attribute Input Data Length Publisher of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Source Offset Publisher
This parameter contains the value of the attribute Source Offset Publisher of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Master Data
This parameter contains the value of the attribute Master Data of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Source Offset Master
This parameter contains the value of the attribute Source Offset Master of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Dest Slot Number
This parameter contains the value of the attribute Dest Slot Number of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Offset Data Area
This parameter contains the value of the attribute Offset Data Area of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Length Data Area
This parameter contains the value of the attribute Length Data Area of one DXB-Subscribertable entry of the DXB-Subscribertable object.

IsoM Parameter
This parameter indicates the presence of the IsoM Parameter.

TBASE_DP
This parameter contains the value of the attribute T_{BASE_DP} of the IsoM Parameter object.

T_{DP}
This parameter contains the value of the attribute T_{DP} of the IsoM Parameter object.

TMAPC
This parameter contains the value of the attribute T_{MAPC} of the IsoM Parameter object.

TBASE_IO
This parameter contains the value of the attribute T_{BASE_IO} of the IsoM Parameter object.

T_I
This parameter contains the value of the attribute T_I of the IsoM Parameter object.

T_O
This parameter contains the value of the attribute T_O of the IsoM Parameter object.

T_{DX}
This parameter contains the value of the attribute T_{DX} of the IsoM Parameter object.

TPLL_W
This parameter contains the value of the attribute T_{PLL_W} of the IsoM Parameter object.

TPLL_D
This parameter contains the value of the attribute T_{PLL_D} of the IsoM Parameter object.

Prm Command
This parameter indicates the presence of the Prm Command.

Seq Number
This parameter contains the Sequence Number necessary for the unique identification of the received Prm Command.

Allowed values: 0 to 31

Primary Request
If this parameter is set to TRUE the selected communication interface shall operate as the primary one. All MS1 functions are active and will be processed as if redundancy is not activated. Also all diagnosis informations including channel related, or other status messages, have to be present.

This parameter set to FALSE means that the role of the communication interface (primary, backup) shall not be changed.

Allowed values: TRUE, FALSE

MS1 Command
This parameter defines the desired command for the MS1 AR of the selected communication interface.

The allowed values are shown in Table 86.

Table 86 – MS1 Command

Value	Meaning
no Action	No Command for the MS1 AR is present.
Stop MS1	This parameter specifies when the MS1 AR of the DP-slave shall be stopped by the DP Application Layer. No MS1 AR services are possible any longer.
Start MS1	This parameter specifies when the MS1 AR of the DP-slave shall be (re)started by the DP Application Layer. After the (re)start all MS1 AR services are possible if the role of the communication interface is primary.
Reset MS1	Means that the MS1 AR shall be stopped and then be started afterwards.

Master State Clear
This parameter indicates the actual state of the MS0 AR of the DP-master (class 1). This parameter set to TRUE means that the MS0 AR of the DP-master (class 1) is in the operation mode CLEAR. This parameter set to FALSE means that the MS0 AR of the DP-master (class 1) is in the operation mode OPERATE.

Allowed values: TRUE, FALSE

Check Properties
This parameter indicates whether the DP-slave shall check the properties provided by the Check User Prm service. The parameter set to TRUE indicates that properties shall be checked (e. g. at the start-up of the DP-slave). This parameter set to FALSE indicates that the check of the properties shall be omitted (e. g. at the data exchange mode of the DP-slave).

Allowed values: TRUE, FALSE

Primary Request used
This parameter indicates whether the communication interface of a redundant DP-slave supports primary requests. The DP-slave shall respond negatively (with the parameter Prm OK of the Check User Prm service set to FALSE) if those requests are not supported. In the other case the Check Properties shall be accepted.

Allowed values: TRUE, FALSE

MS1 Command used
This parameter indicates whether the MS1 Commands Start MS1, Stop MS1 and Reset MS1 are supported by the communication interface of a redundant DP-slave. The DP-slave shall respond negatively (with the parameter Prm OK of the Check User Prm service set to FALSE) if those requests are not supported. In the other case the Check Properties shall be accepted.

Allowed values: TRUE, FALSE

Address Change
This parameter indicates whether address changes are requested with the primary request. TRUE indicates that address changes are requested and FALSE indicates that no address changes are requested. The DP-slave shall respond negatively (with the parameter Prm OK of the Check User Prm service set to FALSE) if those requests are not supported. In the other case the Check Properties shall be accepted.

Allowed values: TRUE, FALSE

Address Offset
This parameter indicates if the address offset of 64 will be used at the start-up of the DP-slave or not. The DP-slave shall respond negatively (with the parameter Prm OK of the Check User Prm service set to FALSE) if the address

offset is not supported. In the other case the Check Properties shall be accepted.

Allowed values: TRUE, FALSE

Output Hold Time
This parameter indicates the time the redundant DP-slave shall not change its outputs when it performs a redundancy switchover. When the backup communication interface becomes the primary (and vice versa) the User may use the output values previously transmitted from the old primary communication interface or the User has to wait for the first data exchange after the switchover to write new values to its outputs. A possible switchover from backup to primary will always start the Output_Hold_Time.

Timebase: 10 ms.

Allowed values: 0 to 2^{16} - 1

Time AR Parameter
This parameter indicates the presence of the Time AR Parameter.

Clock Sync Interval
This parameter contains the value for the attribute Clock Sync Interval of the ARL DP-slave object.

CS Delay Time
This parameter contains the value for the attribute CS Delay Time of the ARL DP-slave object.

6.2.5.3.2 Check user Prm result

This service is used in a DP-slave as a reply to the previous issued Check User Prm service. The Default AP of a DP-slave uses this service to provide the result of the validation of the User Prm Data or of the List of Structured Prm Data. This service shall only be used in conjunction with the MS0 AR.

Table 87 shows the parameters of the service.

Table 87 – Check user Prm result

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Prm OK	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

Prm OK
The parameter Prm OK equal TRUE specifies that the User Prm Data or the List of Structured Prm Data conveyed by the previous Check User Prm service are acceptable by the application of the DP-slave. If they are not acceptable for the application of the DP-slave, this parameter shall have the value FALSE. In this case the attributes Prm Req and Prm Fault of the related MS0 ARL Entry will be set to TRUE by the DP Application Layer.

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

Result(-)
This selection type parameter indicates that the service request failed. The DP-slave shall ignore the DP-masters Parameterization.

Status
This parameter indicates the cause of the failure.

The allowed values are shown in Table 88.

Table 88 – Status values

Value	Meaning
Reset	Reset still in progress
Not Pending	No Validation pending
New Prm	New User Prm Data or List of Structured Prm Data received during validation
Inv Master Add	A different DP-master has parameterized the DP-slave or the DP-master has released the DP-slave (the validation of the previous User Prm Data or List of Structured Prm Data was useless)

6.2.5.3.3 Check Ext user Prm

This service shall only be used if the MS0 AR offers an additional CR for the conveyance of the parameterization data and the device attribute DPV 1 is set to TRUE.

The DP Application Layer of the DP-slave notifies the application if additional parameterization data are received from the DP-master (Class 1). This service requests a validation of the List of Structured Prm Data. A further Check Ext User Prm indication is only issued after the Check Ext User Prm service is completed with a Check Ext User Prm Result service.

The Check Ext User Prm service shall be acknowledged negatively (the parameter Ext Prm OK of the service Check Ext User Prm Result shall be set to FALSE) if at least one parameterization object is unknown or not allowed or acceptable by the application of the DP-slave.

Table 89 shows the parameters of the service.

Table 89 – Check Ext user Prm

Parameter name	Ind
Argument	M
AREP	M
List of Structured Prm Data	M
Structured User Prm Data	U
Device Related User Parameter	U
Device Prm Data	M
Module Related User Parameter	U
Slot Number	M
Module Prm Data	M
DXB-Linktable	U
List of DXB Link Entries	M
Publisher Address	M
Input Data Length Publisher	M
Source Offset Publisher	M
Length	M
DXB-Subscribertable	U
List of DXB Subscriber Entries	M
Publisher Address	C
Input Data Length Publisher	M
Source Offset Publisher	M
Master Data	C
Source Offset Master	M
Dest Slot Number	M
Offset Data Area	M
Length Data Area	M
IsoM Parameter	U
T_{BASE_DP}	M
T_{DP}	M
T_{MAPC}	M
T_{BASE_IO}	M
T_I	M
T_O	M
T_{DX}	M
T_{PLL_W}	M
T_{PLL_D}	M
Time AR Parameter	U
Clock Sync Interval	M
CS Delay Time	U

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the AR.

List of Structured Prm Data
This parameter is composed of one or more of the following list elements.

Structured User Prm Data
This parameter is composed of one or both of the following subparameters.

Device Related User Parameter
This parameter indicates the presence of Device Related User Parameter.

Device Prm Data
This parameter contains the value for the Device Related User Parameter of the MS0 Structured User Parameter object.

Module Related User Parameter
This parameter indicates the presence of Module Related User Parameter.

Slot Number
This parameter contains the value of the attribute Slot Number of the MS0 Structured User Parameter object.

Module Prm Data
This parameter contains the value for the Module Related User Parameter of the MS0 Structured User Parameter object.

DXB-Linktable
This parameter indicates the presence of the DXB-Linktable.

List of DXB Link Entries
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of one DXB-Linktable entry of the DXB-Linktable object.

Input Data Length Publisher
This parameter contains the value of the attribute Input Data Length Publisher of one DXB-Linktable entry of the DXB-Linktable object.

Source Offset Publisher
This parameter contains the value of the attribute Source Offset Publisher of one DXB-Linktable entry of the DXB-Linktable object.

Length
This parameter contains the value of the attribute Length of one DXB-Linktable entry of the DXB-Linktable object.

DXB-Subscribertable
This parameter indicates the presence of the DXB-Subscribertable.

List of DXB Subscriber Entries
This parameter is composed of the following list elements:

Publisher Address
This parameter contains the value of the attribute Publisher Address of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Input Data Length Publisher
This parameter contains the value of the attribute Input Data Length Publisher of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Source Offset Publisher
This parameter contains the value of the attribute Source Offset Publisher of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Master Data
This parameter contains the value of the attribute Master Data of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Source Offset Master
This parameter contains the value of the attribute Source Offset Master of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Dest Slot Number
This parameter contains the value of the attribute Dest Slot Number of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Offset Data Area
This parameter contains the value of the attribute Offset Data Area of one DXB-Subscribertable entry of the DXB-Subscribertable object.

Length Data Area
This parameter contains the value of the attribute Length Data Area of one DXB-Subscribertable entry of the DXB-Subscribertable object.

IsoM Parameter
This parameter indicates the presence of the IsoM Parameter.

TBASE_DP
This parameter contains the value of the attribute T_{BASE_DP} of the IsoM Parameter object.

T_{DP}
This parameter contains the value of the attribute T_{DP} of the IsoM Parameter object.

TMAPC
This parameter contains the value of the attribute T_{MAPC} of the IsoM Parameter object.

TBASE_IO
This parameter contains the value of the attribute T_{BASE_IO} of the IsoM Parameter object.

T_I
This parameter contains the value of the attribute T_I of the IsoM Parameter object.

T_O
This parameter contains the value of the attribute T_O of the IsoM Parameter object.

T_{DX}
This parameter contains the value of the attribute T_{DX} of the IsoM Parameter object.

TPLL_W
This parameter contains the value of the attribute T_{PLL_W} of the IsoM Parameter object.

TPLL_D
This parameter contains the value of the attribute T_{PLL_D} of the IsoM Parameter object.

Time AR Parameter
This parameter indicates the presence of the Time AR Parameter.

Clock Sync Interval
This parameter contains the value for the attribute Clock Sync Interval of the ARL DP-slave object.

CS Delay Time
This parameter contains the value for the attribute CS Delay Time of the ARL DP-slave object.

6.2.5.3.4 Check Ext user Prm result

This service is used in a DP-slave as a reply to the previous issued Check Ext User Prm service. The Default AP of a DP-slave uses this service to provide the result of the validation of the List of Structured Prm Data. This service shall only be used in conjunction with the MS0 AR.

Table 90 shows the parameters of the service.

Table 90 – Check Ext user Prm result

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Ext Prm ~~OK~~	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

Ext Prm OK
The parameter Ext Prm OK equal TRUE specifies that the List of Structured Prm Data conveyed by the previous Check Ext User Prm service are acceptable by the application of the DP-slave. If the List of Structured Prm Data are not acceptable for the application of the DP-slave, this parameter shall have the value FALSE. In this case the attributes Prm Req and Cfg Fault of the related MS0 ARL Entry will be set to TRUE by the DP Application Layer.

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

Result(-)
This selection type parameter indicates that the service request failed. The DP-slave shall ignore the DP-masters Parameterization.

Status
This parameter indicates the cause of the failure.

The allowed values are shown in Table 91.

Table 91 – Status values

Value	Meaning
Reset	Reset still in progress
Not Pending	No Validation pending
New Prm	New List of Structured Prm Data received during validation
Inv Master Add	A different DP-master has parameterized the DP-slave or the DP-master has released the DP-slave (the validation of the previous List of Structured Prm Data was useless).

6.2.5.3.5 Check Cfg

The DP Application Layer of the DP-slave notifies the application when Cfg Data (Configuration Data) are received from the DP-master because this service requests a validation of the Cfg Data by the application. A further Check Cfg indication is only issued after the Check Cfg service is completed with a Check Cfg Result service. The DP-slave compares its own configuration data with the Cfg Data received from the DP-master.

The check shall cover the following details:

— module structure

— range of Input and/or Output Data Element

— format

— consistency.

The consistency also affects the DP-slave and the DP-master. The verification of the consistency in the Cfg Data list element shall cause only in the following cases a configuration fault:

— The DP-slave needs consistency for an Input and/or Output Data Element and the DP-master indicates no consistency.

— The DP-slave can not provide consistency for an Input and/or Output Data Element and the DP-master requires consistency for this Input and/or Output Data Element.

DP-slaves may accept different sets of Cfg Data. In this case the Check Cfg shall be successful if one set of configuration data matches. If the service parameter Check Cfg Mode is set to TRUE the check of the Cfg Data may be done in a different User specific way. For example a temporary not available module may be accepted even if the Cfg Data contains the respective Configuration Identifier.

Table 92 shows the parameters of the service.

Table 92 – Check Cfg

Parameter name	Ind
Argument	—
AREP	M
Check Cfg Mode	C
Cfg Data	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the AR.

Check Cfg Mode
This parameter indicates how the DP-slave application has to check the Cfg Data. This parameter shall only be evaluated if the device attribute DPV1 is set TRUE.

Cfg Data
This parameter contains the Cfg Data to be verified by the application of the DP-slave.

6.2.5.3.6 Check Cfg result

This service is used in a DP-slave as a reply to the previous issued Check Cfg service. The Default AP of the DP-slave uses this service to provide the result of the validation of the Cfg Data. This service shall only be used in conjunction with the MS0 AR.

Table 93 shows the parameters of the service.

Table 93 – Check Cfg result

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Cfg OK	M	
Input Data Len	M	
Output Data Len	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

Cfg OK
The parameter Cfg OK equal TRUE specifies that the Cfg Data conveyed by the previous Check Cfg service are acceptable by the application of the DP-slave. The application of the DP-slave may update the Real Cfg Data of the MS0 AR with the Set Cfg service. If the Cfg Data are not acceptable for the application of the DP-slave, this parameter shall have the value FALSE. In this case the attributes Prm Req and Cfg fault will be set to TRUE by the DP Application Layer.

Input Data Len
This parameter contains the actual length of the Input Data.

Output Data Len
This parameter contains the actual length of the Output Data.

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

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

Status
This parameter indicates the cause of the failure.

The allowed values are shown in Table 94.

Table 94 – Status values

Value	Meaning
Reset	Reset still in progress
Not Pending	No Validation of Cfg Data pending
New Cfg	New Cfg Data received during validation
Inv Master Add	A different DP-master has parameterized the DP-slave or the DP-master has released the DP-slave (the validation of the previous Cfg Data was useless).

6.2.5.3.7 Set Cfg

This service is used by the application of the DP-slave to pass configuration data to the DP-Application Layer. This service shall only be used in conjunction with the MS0 AR.

Table 95 shows the parameters of the service.

Table 95 – Set Cfg

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Cfg Data	M	
Result(+)		S
AREP		M
Result(-)		S
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

Cfg Data
This parameter contains the Cfg Data.

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

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

6.2.5.3.8 Get Cfg

This service allows the DP-master (Class 2) to read the configuration data of the DP-slave via the MS0 AR.

Table 96 shows the parameters of the service.

Table 96 – Get Cfg

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result(+)		S
AREP		M
Cfg Data		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

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

Cfg Data
This parameter contains the Cfg Data.

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

Status
The parameter Status indicates the type of failure of the service.

Allowed values: DS, NA, RS, UE, NR, RE

6.2.5.3.9 Set slave add

This service permits a DP-master (Class 2) to change the address of a DP-slave and to pass remanent Slave Data. The Set Slave Add service is only successful if the Ident Number transmitted corresponds with the Ident Number of the DP-slave and the attribute No Add Change (see 6.2.10.3.1) of the DP-slave has the value FALSE. This service shall only be used in conjunction with the MS0 AR.

Table 97 shows the parameters of the service.

Table 97 – Set Slave Add

Parameter name	Req	Ind	Cnf
Argument	M	M(=)	
AREP	M	M(=)	
New Slave Add	M	M(=)	
Ident Number	M	M(=)	
No Add Chg	M	M(=)	
Rem Slave Data	U	C(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

New Slave Add
This parameter contains the value of the attribute New Slave Add of the Remanent Parameter object of the DP-slave which is to be set.

Ident Number
This parameter contains the Ident Number to be checked with the Ident Number of the DP-slave.

No Add Chg
This parameter contains the value of the attribute No Add Change of the Remanent Parameter object. The attribute No Add Change will assume the value of this service parameter in case of success, otherwise No Add Change remains unchanged.

Rem Slave Data
This parameter contains the value of the attribute Rem Para of the Remanent Parameter object.

Result(+)
This selection type parameter indicates that the service request succeeded. This confirmation does not reflect whether the service was successful, i. e. the new values were accepted by the DP-slave. Therefore the DP-master shall check the correct execution of the service by using the Read Slave Diag service with the new DP-slave address.

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

Status
The parameter Status indicates the type of failure of the service.

Allowed values: DS, NA, RS, RR, UE, RE

6.2.5.3.10 Initiate

This service shall be used by a DP-master (Class 2) to establish a connection to a desired DP-slave. This service shall only be used in conjunction with a MS2 AR.

Table 98 shows the parameters of the service.

Table 98 – Initiate

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Send Timeout	M			
Features Supported	M	M(=)		
Profile Features Supported	M	M(=)		
Profile Ident Number	M	M(=)		
Add Addr Param	M	M(=)		
S-Type	M	M(=)		
S-Len	M	M(=)		
D-Type	M	M(=)		
D-Len	M	M(=)		
S-Addr	M	M(=)		
S-API	M	M(=)		
S-SCL	M	M(=)		
S-Network Address	C	C(=)		
S-MAC Address	C	C(=)		
D-Addr	M	M(=)		
D-API	M	M(=)		
D-SCL	M	M(=)		
D-Network Address	C	C(=)		
D-MAC Address	C	C(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Max Len Data Unit			M	M(=)
Features Supported			M	M(=)
Profile Features Supported			M	M(=)
Profile Ident Number			M	M(=)
Add Addr Param			M	M(=)
S-Type			M	M(=)
S-Len			M	M(=)
D-Type			M	M(=)
D-Len			M	M(=)
S-Addr			M	M(=)
S-API			M	M(=)
S-SCL			M	M(=)
S-Network Address			C	C(=)
S-MAC Address			C	C(=)

Parameter name	Req	Ind	Rsp	Cnf
D-Addr			M	M(=)
D-API			M	M(=)
D-SCL			M	M(=)
D-Network Address			C	C(=)
D-MAC Address			C	C(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Decode			M	M(=)
Error Code 1			M	M(=)
Error Code 2			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Max Len Data Unit
This parameter contains the value of the attribute Max Len Data Unit of the ARL DP-slave object of a DP-slave.

Send Timeout
This parameter defines the control time for monitoring the MS2 AR requested by the DP-master (Class 2). DP-master and DP-slave inform each other about the supported Send Timeout. The DP-slave adjust its Send Timeout. The DP-master (Class 2) decides whether the reported Send Timeout is sufficient or not.

Features Supported
DP-master and DP-slave inform each other about the supported service functionality. The DP-slave has the possibility to adjust its functionality to the Master's requirements. The User application of the DP-master has to decide if the reported functionality is sufficient or not.

Profile Features Supported
DP-master and DP-slave inform each other about the supported service functionality regarding the used profile definition. The profile is identified by the Profile Ident Number. The meaning of the defined bits is profile or vendor specific.

Profile Ident Number
This parameter identifies a profile definition uniquely. All devices using the same profile definition have to use the same Profile Ident Number. The Profile Ident Number will be taken from the pool of Ident Numbers for vendor specific or authorized profiles. The value 0 indicates that no profile is supported. If the requested profile is supported by the DP-slave, the Profile Ident Number is mirrored in the response. If the requested profile is not supported by the DP-slave, the DP-slave responds negatively or with a Profile Ident Number supported by the DP-slave.

Add Addr Param
Add Addr Param consists of two parts, additional address information of the source and additional address information of the destination. The additional address information contains the parameter API plus an optional Network/MAC address.

D-Addr and S-Addr (API, SCL, Network Address, MAC Address) are related to the direction of the telegrams, which means the source becomes destination and vice versa in the response.

Add Addr Param contains the following subparameters:

S-Type
This subparameter indicates the presence (S-Type = 1) of the optional Network/MAC address in the Add Addr Param of the source.

S-Len
This subparameter indicates the length of the S-Addr subparameter.

D-Type
This subparameter indicates the presence (D-Type = 1) of the optional Network/MAC address in the Add Addr Param of the destination.

D-Len
This subparameter indicates the length of the D-Addr subparameter.

S-Addr
This subparameter contains the additional address information of the source:

S-API
This subparameter identifies the AP of the source.

S-SCL
This subparameter identifies the access level of the source.

S-Network Address
In case of S-Type = 1 this subparameter identifies the network address of the source according to ISO/OSI-Network addresses.

S-MAC Address
In case of S-Type = 1 this subparameter identifies the MAC Address of the source.

D-Addr
This subparameter contains the additional address information of the destination:

D-API
This subparameter identifies the AP of the destination.

D-SCL
This subparameter identifies the access level of the destination.

D-Network Address
In case of D-Type = 1 this subparameter identifies the network address of the destination according to ISO/OSI-Network addresses.

D-MAC Address
In case of D-Type = 1 this subparameter identifies the MAC Address of the destination.

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

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

Error Decode
This parameter selects one of the following Error schemes: DPV1, FMS, Profile_Specific.

Error Code 1
In case of Error Decode equal DPV1 the Error Code 1 assumes one of the following values:
version conflict, feature not supported, user specific, state conflict, invalid range, invalid parameter, invalid type, resource busy, resource unavailable.

Error Code 2
The parameter Error Code 2 is user specific.

6.2.5.3.11 Abort

The DP-master and DP-slave use this service to abort an AR. With the Abort.req the User resets the FSPM. In faulty conditions the FSPM informs the User with an Abort.ind and resets itself. This service shall only be used in conjunction with the MS2 AR.

Table 99 shows the parameters of the service.

Table 99 – Abort

Parameter name	Req	Ind
Argument	M	M
AREP	M	M(=)
Locally Generated		C
Subnet	C	C(=)
Instance	C	C(=)
Reason Code	C	C(=)
Additional Detail		C

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is used as local identifier of the desired AR.

Locally Generated
This parameter only exists in the indication in case of an MS2 AR. It indicates that the Abort has been produced by the remote partner (Locally Generated = FALSE) or in the local station (Locally Generated = TRUE).

Subnet
This parameter is used in case of an MS2 AR to encode information about the location of the source of the Abort.

Instance
This parameter is used in case of an MS2 AR to encode information about the protocol instance that detected a problem causing the Abort.

The allowed values are shown in Table 100.

Table 100 – Instance

Value	Meaning
DL	Data-link layer
MS2	MS2 AR
User	User

Reason Code
The Reason Code indicates the reason for the Abort in case of an MS2 AR.

Allowed Values: UE, RR, RS, NR, DH, LR, RDL, RDH, DS, NA, ABT_SE, ABT_FE, ABT_TO, ABT_RE, ABT_IV, ABT_STO, ABT_IA, ABT_OC.

Additional Detail
This parameter contains detail information in case of an abort which is caused by an inappropriate Initiate (e.g. Send Timeout to small) in case of an MS2 AR. This parameter is only present if the parameter Reason Code has the value ABT_STO and the parameter Instance equals MS2. In this case the parameter additional detail contains the value of the received Send Timeout of the DP-slave.

6.2.5.3.12 MS0 init DP-slave

This service is used to initialize the MS0 AR of a DP-slave.

Table 101 shows the parameters of the service.

Table 101 – MS0 init DP-slave

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result		M
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Result
This parameter indicates that the service request succeeded.

6.2.5.3.13 MS1 init DP-slave

This service is used to initialize the MS1 AR of a DP-slave.

Table 102 shows the parameters of the service.

Table 102 – MS1 init DP-slave

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result		M
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Result
This parameter indicates that the service request succeeded.

6.2.5.3.14 MS2 init DP-slave

This service is used to initialize all MS2 AR of a DP-slave.

Table 103 shows the parameters of the service.

Table 103 – MS2 init DP-slave

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result		M
AREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Result
This parameter indicates that the service request succeeded.

6.2.5.3.15 DP-slave started

With this service the supported alarm features on this MS1 AR are indicated. The indication occurs after the DP-slave has entered the data exchange mode and the transfer of alarm and status messages is possible from now on.

Table 104 shows the parameters of the service.

Table 104 – DP-slave started

Parameter name	Ind
Argument	M
AREP	M
Actual Enabled Alarms	M
Alarm Sequence	M
Alarm Limit	C

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Actual Enabled Alarms
This parameter indicates which alarm types are supported by the DP-master (Class 1).

Allowed values: Update Alarm enabled / disabled, Manufacturer Specific Alarm enabled / disabled, Diagnosis Alarm enabled / disabled, Process Alarm enabled / disabled, Pull Plug Alarm enabled / disabled

Alarm Sequence
This Boolean parameter indicates whether only one alarm of a specific Alarm Type can be active at one time (Alarm Sequence=FALSE) or several alarms (2 to 32) of the same or different type can be active at one time (Alarm Sequence=TRUE).

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Alarm Limit
This parameter indicates the maximum number of parallel alarms.

The allowed values are shown in Table 105.

Table 105 – Alarm limit

Value	Meaning
0	1 alarm of each type
1	2 alarms in total
2	4 alarms in total
3	8 alarms in total
4	12 alarms in total
5	16 alarms in total
6	24 alarms in total
7	32 alarms in total

6.2.5.3.16 DP-slave stopped

In case of a an MS0 AR, this Application Layer service indicates to the User that the DP-slave has left the data exchange mode and that the MS1 AR is closed. As a consequence the transfer of alarms as well as the processing of Write and Read services is no longer possible.

In case of a an MS1 AR, this service indicates to the User that the acyclic communication relationship to the DP-master (Class 1) has been stopped. This means that there is no longer cyclic data exchange, the MS0 AR and MS1 AR of a DP-slave will be set to the initial state. Table 106 shows the parameters of the service.

Table 106 – DP-slave stopped

Parameter name	Ind
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

6.2.5.3.17 Reset DP-slave

By this service the User of a DP-slave resets the Application Layer.

Table 107 shows the parameters of the service.

Table 107 – Reset DP-slave

Parameter name	Req	Cnf
Argument		

Argument
The argument shall convey the service specific parameters of the service request. Nevertheless, in this service there are no specific parameters.

6.2.5.3.18 DP-slave fault

By this service the Application Layer indicates to the User of a DP-slave that due to fatal errors all ARs have been reset.

Table 108 shows the parameters of the service.

Table 108 – DP-slave fault

Parameter name	Ind
Argument	

Argument
The argument shall convey the service specific parameters of the service request. Nevertheless, in this service there are no specific parameters.

6.2.5.3.19 Application ready DP-slave

By this service the User of a DP-slave indicates to the Application Layer that the application is ready. This service shall only be used in conjunction with the MS0 AR.

Table 109 shows the parameters of the service.

Table 109 – Application ready DP-slave

Parameter name	Req
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

6.2.5.3.20 Start subscriber

This service is used by the application of the DP-slave to start the subscription of a certain DXB-Link.

Table 110 shows the parameters of the service.

Table 110 – Start subscriber

Parameter name	Req	Cnf
Argument	M	
AREP	M	
CREP	M	
Result(+)		S
AREP		M
CREP		M
Result(-)		S
AREP		M
CREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
This parameter is the local identifier for the desired CR.

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

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

6.2.5.3.21 Stop subscriber

This service is used by the application of the DP-slave to stop the subscription of a certain DXB-Link.

Table 111 shows the parameters of the service.

Table 111 – Stop subscriber

Parameter name	Req	Cnf
Argument	M	
AREP	M	
CREP	M	
Result(+)		S
AREP		M
CREP		M
Result(-)		S
AREP		M
CREP		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
This parameter is the local identifier for the desired CR.

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

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

6.2.5.3.22 Publisher active

This service indicates to the application of the Subscriber (DP-slave) the state of the Publisher of a certain DXB-Link. This service will be issued by the DP Application Layer only if the state of the Publisher has been changed since the last watchdog period.

Table 112 shows the parameters of the service.

Table 112 – Publisher active

Parameter name	Ind
Argument	M
AREP	M
CREP	M
Status	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

CREP
This parameter is the local identifier for the desired CR (Publisher).

Status
This parameter indicates the state of the related Publisher.

The allowed values are shown in Table 113.

Table 113 – Status

Value	Meaning
TRUE	the related Publisher is active: during the last watchdog period the subscribed Input Data object of the related Publisher has been received at least one time
FALSE	the related Publisher is not active; during the last watchdog the subscribed Input Data object of the related Publisher has not been received due to the following problems: the value of the attribute Input Data Length Publisher of the related Publisher of the related CRL entry was not in alignment with the actual length of the Input Data object published by the related Publisher the related Publisher was not in the data exchange mode breakdown of the related Publisher

6.2.5.3.23 Init DP-master CI1

By this service the User initializes the Application Layer of a DP-master (Class 1). The parameter are described in this specification, part 3 and 4.

Table 114 shows the parameters of the service.

Table 114 – Init DP-master CI1

Parameter name	Req	Cnf
Argument	M	
Bus Para	M	
TS	M	
Data rate	M	
TSL	M	
minT_{SDR}	M	
maxT_{SDR}	M	
TQUI	M	
TSET	M	
TTR	M	
G	M	
HSA	M	
max retry limit	M	
Result		M

Argument
The argument shall convey the service specific parameters of the service request.

Bus Para
This parameter is composed of the following subparameters:

TS
This parameter contains the DL-address of this DP-master (Class 1).

Data rate
This parameter contains the data rate of this DP-master (Class 1).

TSL
This parameter contains the slot time for this DP-master (Class 1).

$minT_{SDR}$
This parameter contains the smallest station delay time for this DP-master (Class 1).

$maxT_{SDR}$
This parameter contains the largest station delay time for this DP-master (Class 1).

TQUI
This parameter contains the transmitter fall time for this DP-master (Class 1).

TSET
This parameter contains the setup time for this DP-master (Class 1).

TTR
This parameter contains the target rotation time for this DP-master (Class 1).

G
This parameter contains the GAP update factor for this DP-master (Class 1).

HSA
This parameter contains the highest station (DL-) address for this DP-master (Class 1).

max retry limit
This parameter contains the maximum number of retries for this DP-master (Class 1).

Result
This parameter indicates that the service request succeeded.

6.2.5.3.24 DP-master CI1 started

In case of a MS0 AR, this Application Layer service indicates to the User of a DP-master (Class 1) that the assigned DP-slave is in the data exchange mode.

In case of a MS1 AR, this Application Layer service indicates that the handling of alarm requests for this DP-slave are possible.

Table 115 shows the parameters of the service.

Table 115 – DP-master CI1 started

Parameter name	Ind
Argument	M
AREP	M
Alarm Limit	C

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR (Address of DP-slave).

Alarm Limit
In case of a an MS1 AR, this parameter indicates the maximum number of parallel alarms.

The allowed values are shown in Table 116.

Table 116 – Alarm limit

Value	Meaning
0	1 alarm of each type
1	2 alarms in total
2	4 alarms in total
3	8 alarms in total
4	12 alarms in total
5	16 alarms in total
6	24 alarms in total
7	32 alarms in total

6.2.5.3.25 DP-master Cl1 stopped

This Application Layer service indicates to the User of a DP-master (Class 1) that the related AR has been stopped. In case of a MS0 AR, the MS1 AR has been stopped, too. A new start is necessary to restart MS0 AR and MS1 AR.

Table 117 shows the parameters of the service.

Table 117 – DP-master Cl1 stopped

Parameter name	Ind
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

6.2.5.3.26 Reset DP-master Cl1

By this service the User of a DP-master (Class 1) resets the Application Layer.

Table 118 shows the parameters of the service.

Table 118 – Reset DP-master Cl1

Parameter name	Req	Cnf
Argument	M	
Result		M

Argument
The argument shall convey the service specific parameters of the service request. Nevertheless, in this service there are no specific parameters.

Result
This parameter indicates that the service request succeeded.

6.2.5.3.27 DP-master CI1 fault

By this service the Application Layer indicates to the User of a DP-master (Class 1) that a fatal error has occurred and all ARs have been stopped. Further communication is only possible after a Reset.req of the User.

Table 119 shows the parameters of the service.

Table 119 – DP-master CI1 fault

Parameter name	Ind
Argument	—

Argument
The argument shall convey the service specific parameters of the service request. Nevertheless, in this service there are no specific parameters.

6.2.5.3.28 DP-master CI1 reject

By this service the Application Layer rejects a previous Read/Write.req from the User of a DP-master (Class 1) to the MS1 AR, if the MS1 AR was forced to stop.

Table 120 shows the parameters of the service.

Table 120 – DP-master CI1 reject

Parameter name	Ind
Argument	M
AREP	M
Reason	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR (Address of DP-slave).

Reason
The parameter Status indicates success or failure of the service.

Allowed values: REJ ABORT,REJ LE,REJ PS,REJ SE,REJ IV

6.2.5.3.29 Set mode DP-master CI1

By this service the User of a DP-master (Class 1) can set the operation mode of the MS0 AR. Additionally the User gets information about the actual state of the bus system in case of errors.

Table 121 shows the parameters of the service.

Table 121 – Set mode DP-master CI1

Parameter name	Req	Cnf
Argument	M	
AREP	M	
USIF State	M	
Result(+)		M
AREP		M
Bus Accessible		M
Result(-)		
AREP		M
Bus Accessible		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

USIF State
The Application Layer provides four operation modes for the User:

OFFLINE
Communication to all DP participants (DP-master and DP-slave) is stopped. MAC of local DL is in offline state. The Application Layer waits for a signal to start.

STOP
Changing the operation mode from OFFLINE to STOP the bus parameter set is loaded in the DL by the Application Layer. The DL was activated. The responder services for MM1 AR and MM2 AR can be processed. MS0 and MS1 AR will not be activated (No interactions to DP-slaves).

CLEAR
In this operation mode the DP-master (Class 1) attempts to configure the DP-slaves that are assigned to it, and starts the data exchange mode. The input data of the DP-slaves are read and passed to the User. The User's output data are ignored. The assigned DP-slaves receive a value of 0 or no data for their Output Data objects. (depending on the attribute Fail Safe)

OPERATE
The DP-master (Class 1) is in the data exchange mode with the assigned DP-slaves. The inputs from the DP-slaves are transferred to the User and the outputs from the User are passed to the DP-slaves. If the Application Layer leaves the mode OPERATE, all DP-slaves' outputs change in the safe state by means of the service Global Control (Control Command = Clear Data, Group Select = 0).

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

Bus Accessible
This Boolean parameter indicates whether the DP-master is able to access the bus (TRUE) or not (FALSE).

Attribute Type: Boolean

Allowed values: TRUE, FALSE

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

6.2.5.3.30 DP-master CI1 mode changed

By this service the Application Layer indicates to the user of a DP-master (Class 1):

— the change of its operation mode from OPERATE to CLEAR if an error occurred and the Error Action Flag is set

— the change of its operation mode from STOP to OFFLINE if an error occurred while setting the bus parameters.

Table 122 shows the parameters of the service.

Table 122 – DP-master CI1 mode changed

Parameter name	Ind
Argument	M
AREP	M
USIF State	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

USIF State
Allowed values: OFFLINE, CLEAR (see 6.2.5.3.29)

6.2.5.3.31 Load bus Par DP-master CI1

By this service the User of a DP-master (Class 1) can load a new bus parameter set. This service is not allowed in the operation mode OFFLINE. During STOP changes to all bus parameters are possible. In the operation modes CLEAR and OPERATE of the Application Layer it is not allowed to load a bus parameter set with changes to critical parameters DL variables Data_rate and TS.

The changed bus parameters are not stored. Thus, all parameters of this service are lost after changing the operation mode of the Application Layer from STOP to OFFLINE. Changes to be stored can only be done within the bus parameter set (Bus Para) of the Init service.

Table 123 shows the parameters of the service.

Table 123 – Load bus Par DP-master CI1

Parameter name	Req	Cnf
Argument	M	
Bus Para	M	
Result(+)		S
Status		M
Result(-)		S
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Bus Para
This parameter contains the bus parameter set which shall be loaded. The structure is defined in Master CI1 Init service.

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

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

Status
The parameter Status indicates the reason for success or failure of the service.

Allowed values: NO, IV

6.2.5.3.32 Mark DP-master CI1

By this service the User of a DP-master (Class 1) can operate synchronously with the MS0 AR. After the User has sent this service to the Application Layer, the User will receive an acknowledgement from the Application Layer after all activated DP-slaves were polled at least once.

Table 124 shows the parameters of the service.

Table 124 – Mark DP-master CI1

Parameter name	Req	Cnf
Argument	M	
AREP	M	
Result(+)		S
AREP		M
Dia		M
Result(-)		S
AREP		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

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

Dia
This parameter indicates (if the value is equal TRUE) whether the DP-master has not executed a data exchange sequence to at least one activated DP-slave.

Allowed values: TRUE, FALSE

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

Status
The parameter Status indicates success or failure of the service. The value NO will be signaled if the Application Layer is in the operation mode OFFLINE or STOP, or a previous Mark.req is still active.

Allowed values: NO

6.2.5.3.33 Abort DP-master CI1

This service allows to release a MS0, MS1 or MM1 AR of a DP-master (Class 1). The indication may be issued by the MS1 AR if the DP-slave responds incorrectly. In this case the MS0 CRs to that DP-slaves will also be released.

Table 125 shows the parameters of the service.

Table 125 – Abort DP-master CI1

Parameter name	Req	Cnf
Argument	M	M
AREP	M	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

6.2.5.3.34 Read value DP-master Cl1

The User of a DP-master (Class 1) read any of the DL variables with this service.

Table 126 shows the parameters of the service.

Table 126 – Read value DP-master Cl1

Parameter name	Req	Cnf
Argument	M	
Variable	M	
Result		M
Value		M
Status		M

Argument
The argument shall convey the service specific parameters of the service request.

Variable
The parameter Variable selects the DL Variable whose current value has to be read, see IEC 61158-3-3 for a list of all possible DL variables.

Result
This parameter indicates that the service request succeeded.

Value
This parameter contains the actual value of the read DL Variable.

Status
The parameter Status indicates success or failure of the service.

Allowed values: OK, NO, IV

6.2.5.3.35 Delete SC DP-master Cl1

By this service the DL statistic counters can be cleared (set to zero) by the User of a DP-master (Class 1).

Table 127 shows the parameters of the service.

Table 127 – Delete SC DP-master Cl1

Parameter name	Req	Cnf
Argument	M	
Address	M	
Result		M

Argument
The argument shall convey the service specific parameters of the service request.

Address
This parameter specifies the DL-address of the statistic counters which have to be cleared.

Attribute Type Unsigend8

Allowed values: 0 to 125

Result
This parameter indicates that the service request succeeded.

6.2.5.3.36 DP-master CI1 event

By this service the events of the service DLM-Event are indicated to the User of a DP-master (Class 1) as an Event.

Table 128 shows the parameters of the service.

Table 128 – DP-master CI1 event

Parameter name	Ind
Argument	M
Event	M
Add Info	M

Argument
The argument shall convey the service specific parameters of the service request.

Event
This parameter can take the values which are allowed for the DLM-Event service.

Add Info
The parameter indicates the added info of the DL.

6.2.5.3.37 Init DP-master CI2

By this service the User initializes the DL-parameter of a DP-master (Class 2).

Table 129 shows the parameters of the service.

Table 129 – Init DP-master CI2

Parameter name	Req	Cnf
Argument	M	
Bus Para	M	
TS	M	
Data rate	M	
TSL	M	
$minT_{SDR}$	M	
$maxT_{SDR}$	M	
TQUI	M	
TSET	M	
TTR	M	
G	M	
HSA	M	
max retry limit	M	
Result		M

Argument
The argument shall convey the service specific parameters of the service request.

Bus Para
This parameter is composed of the following subparameters:

TS
This parameter contains the DL-address of this DP-master (Class 2).

Data rate
This parameter contains the data rate of this DP-master (Class 2).

TSL
This parameter contains the slot time for this DP-master (Class 2).

$minT_{SDR}$
This parameter contains the smallest station delay time for this DP-master (Class 2).

$maxT_{SDR}$
This parameter contains the largest station delay time for this DP-master (Class 2).

TQUI
This parameter contains the transmitter fall time for this DP-master (Class 2).

TSET
This parameter contains the setup time for this DP-master (Class 2).

TTR
This parameter contains the target rotation time for this DP-master (Class 2).

G
This parameter contains the GAP update factor for this DP-master (Class 2).

HSA
This parameter contains the highest station (DL-) address for this DP-master (Class 2).

max retry limit
This parameter contains the maximum number of retries for this DP-master (Class 2).

Result

This parameter indicates that the service request succeeded.

6.2.5.3.38 Reset DP-master CI2

By this service the User of a DP-master (Class 2) resets the Application Layer.

Table 130 shows the parameters of the service.

Table 130 – Reset DP-master CI2

Parameter name	Req	Cnf
Argument		

Argument

The argument shall convey the service specific parameters of the service request. Nevertheless, in this service there are no specific parameters.

6.2.5.3.39 DP-master CI2 fault

By this service the Application Layer indicates a fatal DL error to the User of a DP-master (Class 2).

Table 131 shows the parameters of the service.

Table 131 – DP-master CI2 fault

Parameter name	Ind
Argument	

Argument

The argument shall convey the service specific parameters of the service request. Nevertheless, in this service there are no specific parameters.

6.2.5.3.40 DP-master CI2 reject

By this service the Application Layer rejects a previous Read, Write, Data Transport service from the User of a DP-master (Class 2).

Table 132 shows the parameters of the service.

Table 132 – DP-master CI2 reject

Parameter name	Ind
Argument	M
AREP	M
Reason	M

Argument

The argument shall convey the service specific parameters of the service request.

AREP

This parameter is the local identifier for the desired AR (Address of DP-slave).

Reason
The parameter Status indicates success or failure of the service.

Allowed values: REJ ABORT,REJ LE,REJ PS,REJ SE,REJ IV

6.2.5.3.41 DP-master CI2 closed

By this service the Application Layer informs the User of a DP-master (Class 2) that the MS2 AR has been closed.

Table 133 shows the parameters of the service.

Table 133 – DP-master CI2 closed

Parameter name	Ind
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

6.2.5.3.42 DP-master CI2 event

By this service the events of the DLM are indicated to the User of a DP-master (Class 2) as an Event.

Table 134 shows the parameters of the service.

Table 134 – DP-master CI2 event

Parameter name	Ind
Argument	M
Event	M
Add Info	M

Argument
The argument shall convey the service specific parameters of the service request.

Event
This parameter can take the values which are allowed for the DLM-Event service.

Add Info
The parameter indicates the added info of the DL.

6.2.6 Management ASE

6.2.6.1 Overview

The Management ASE is used to convey parameters between a DP-master (Class 2) and a DP-master (Class 1). Additionally, the DP-master (Class 2) can read diagnosis information of the DP-master (Class 1) and its assigned DP-slaves. The access of the DP-master (Class 2) to the DP-master (Class 1) is performed according to the Client/Server access model.

6.2.6.2 Management class specification

6.2.6.2.1 Master diag class specification

6.2.6.2.1.1 Template

A Master Diag object is described by the following template:

DP ASE:			**Management ASE**
CLASS:		**Master Diag**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Master Status
2.1.	(m)	Attribute:	USIF State
2.2.	(m)	Attribute:	Ident Number
2.3.	(m)	Attribute:	Hardware Release DP
2.4.	(m)	Attribute:	Firmware Release DP
2.5.	(m)	Attribute:	Hardware Release User
2.6.	(m)	Attribute:	Firmware Release User
3.	(m)	Attribute:	System Diagnosis
4.	(m)	Attribute:	Data Transfer List
5.	(o)	Attribute:	List of DP-slave Diag
5.1	(m)	Attribute:	DP-slave Address
5.2	(m)	Attribute:	DP-slave Diag
SERVICES:			
1.	(o)	OpsService:	Get Master Diag

6.2.6.2.1.2 Attributes

Implicit
The attribute Implicit indicates that the Master Diag object is implicitly addressed by the service.

Master Status
This parameter is composed of the following items.

USIF State
This attribute describes the state of a DP-master (Class 1)

The allowed values are shown in Table 135.

Table 135 – USIF state

Value	Meaning
0	OFFLINE
1	STOP In this state the DP-master (Class 1) does not execute any services at the MS0 and MS1 AR. The MM1 and MM2 AR can be established.
2	CLEAR The DP-master is operating the MS0 AR. The Output Data are in the safe state. The services related with MS1 AR can be executed. The MM1 and MM2 AR can be established.
3	OPERATE The DP-master is operating the MS0 AR. The Output Data are valid. The services related with MS1 AR can be executed. The MM1 and MM2 AR can be established.

Ident Number
This attribute reflects the Ident Number of the DP-master (Class 1).

Attribute Type: Unsigned16

Hardware Release DP
This attribute specifies the hardware release of the DP implementation of the DP-master (Class 1).

Attribute Type: Unsigned8

Firmware Release DP
This attribute specifies the firmware/software release of the DP implementation of the DP-master (Class 1).

Attribute Type: Unsigned8

Hardware Release User
This attribute specifies the hardware release of the User implementation of the DP-master (Class 1).

Attribute Type: Unsigned8

Firmware Release User
This attribute specifies the firmware/software release of the User implementation of the DP-master (Class 1).

Attribute Type: Unsigned8

System Diagnosis
The System Diagnosis reflects the state of all DP-slaves assigned to the DP-master (Class 1). This state itself is a one-bit-information for each DP-slave which is set to the value 1 if its Ext Diag Flag of the Diagnosis information is set or if it does not respond or if it responds incorrectly so that the MS0 AR has to be re-established. The state of a DP-slave is reset to the value 0 after the MS0 AR is successfully re-established or if the DP-slave is deactivated by the AP of the DP-master (Class 1) or the USIF State of the DP-master (Class 1) is STOP.

Attribute Type: List of 0,1

Data Transfer List
The Data Transfer List reflects the status of the data exchange mode of all DP-slaves assigned to the DP-master (Class 1).

This status itself is a one-bit-information for each DP-slave which is set to the value 1 if the data exchange with the assigned DP-slave was successfully or if between two data exchange requests is only one diagnosis request. All other cases reset the status bit-information of the DP-slave to the value 0.

Attribute Type: List of 0,1

List of DP-slave Diag
This attribute is composed of the following list elements.

DP-slave Address
This attribute specifies the DL-address of the DP-slave.

Attribute Type: Unsigned8

Allowed values: 0 to 125

DP-slave Diag
This attribute reflects the diagnosis information of the DP-slave inclusive the information added by the DP Application Layers for each DP-slave.

6.2.6.2.1.3 Invocation of master diag object

For the invocation of the Master Diag object the following rules apply:

Only one Master Diag object shall be invoked in a DP-master (Class 1)

6.2.6.2.2 Master parameter class specification

6.2.6.2.2.1 Overview specification

The Master Parameter object consists of one Bus Para element and one or more Slave Para elements. The values for the Master Parameter object may be loaded via the MM1 AR.

The Download of the attribute values for Master Parameter object may be done in one sequence or the attribute values for of the Bus Para element and for the Slave Para elements may be loaded separately.

For the download of the attribute values of Master Parameter object the Application Layer has to be in the operation mode STOP. The operation mode of the Application Layer of the DP-master (Class 1) shall be changed with the Act Param service via the MM1 AR in conjunction with the local service Set Mode DP-master CI1.

If the attribute values for Master Parameter object are loaded in one sequence the values have to be stored first in an intermediate buffer. After reception of the Act Param or Act Para Brct service the User has to transfer the values from the intermediate buffer to the attributes of the Master Parameter object. Afterwards the User shall transfer these values accordingly with the Load ARL, Load CRL (parameter Update set to FALSE) and Load Bus Par DP-master CI1 services. The Act Param service shall be confirmed according to the following rules:

— positive, if all load services are confirmed positively

— negative, if one or more load services are confirmed negatively.

If the attribute values for the Bus Para elements are loaded the values have to be stored first in an intermediate buffer. After reception of the Act Param or Act Para Brct service the User has to transfer the values from the intermediate buffer with the Load Bus Par DP-master CI1 service to the DP Application Layer. The Act Param service shall be confirmed positively, if the previously requested load service are confirmed positively and after the values has been transferred to the attributes of the Master Parameter object.

The Act Param service shall be confirmed negatively, if the previously requested load service are confirmed negatively. In this case the previously transmitted values for the Bus Para elements shall be omitted.

If the attribute values for one Slave Para element are loaded the values have to be stored first in an intermediate buffer. After reception of the Act Param service the User has to transfer the values from the intermediate buffer to the attributes of the Master Parameter object.

Afterwards the User shall transfer these values accordingly with the ARL Slave Update DP-master CI1 and the Load CRL (parameter Update set to TRUE) services. The Act Param service shall be confirmed according to the following rules:

— positive, if all load services are confirmed positively,

— negative, if one or more load services are confirmed negatively.

Care shall be taken by the User because updating of one Slave Para element may interfere with already loaded Bus Para and may change the system behavior.

6.2.6.2.2.2 Template

A Master Parameter object is described by the following template:

DP ASE:			**Management ASE**
CLASS:		**Master Parameter**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Bus Para
2.1.	(m)	Attribute:	Bus Para Len
2.2.	(m)	Attribute:	TS
2.3.	(m)	Attribute:	Data rate
2.4.	(m)	Attribute:	T_{SL}
2.5.	(m)	Attribute:	min T_{SDR}
2.6.	(m)	Attribute:	max T_{SDR}
2.7.	(m)	Attribute:	T_{QUI}
2.8.	(m)	Attribute:	T_{SET}
2.9.	(m)	Attribute:	T_{TR}
2.10.	(m)	Attribute:	G
2.11.	(m)	Attribute:	HSA
2.12.	(m)	Attribute:	max retry limit
2.13.	(m)	Attribute:	Error Action Flag
2.14.	(o)	Attribute:	Isochronous Mode
2.15.	(o)	Attribute:	IsoM Freeze
2.16.	(o)	Attribute:	IsoM Sync
2.17.	(m)	Attribute:	Min Slave Interval
2.18.	(m)	Attribute:	Poll Timeout
2.19.	(m)	Attribute:	Data Control Time
2.20.	(o)	Attribute:	Alarm Max
2.21.	(m)	Attribute:	Max User Global Control
2.22.	(m)	Attribute:	Master User Data Len
2.23.	(m)	Attribute:	Master Class2 Name
2.24.	(m)	Attribute:	Master User Data
2.25.	(o)	Attribute:	T_{CT}
2.26.	(o)	Attribute:	maxT_{SH}
3.	(o)	Attribute:	List of Slave Para
3.1.	(m)	Attribute:	Slave DL-address
3.2.	(m)	Attribute:	Slave Para Len
3.3.	(o)	Attribute:	Extra Alarm SAP
3.4.	(o)	Attribute:	DPV1 Data Types
3.5.	(o)	Attribute:	DPV1 Supported
3.6.	(m)	Attribute:	Fail Safe
3.7.	(m)	Attribute:	Active
3.8.	(m)	Attribute:	Slave Type
3.9.	(m)	Attribute:	New Prm
3.10.	(m)	Attribute:	Max Diag Data Len
3.11.	(o)	Attribute:	Max Channel Data Length
3.12.	(m)	Attribute:	Diag Upd Delay
3.13.	(o)	Attribute:	Alarm Mode
3.14.	(m)	Attribute:	NA to Abort
3.15.	(m)	Attribute:	Ignore AClr
3.16.	(o)	Attribute:	Publisher Flag
3.17.	(o)	Attribute:	MS1 Timeout
3.18.	(o)	Attribute:	Prm Command supported
3.19.	(m)	Attribute:	Prm Data Len
3.20.	(m)	Attribute:	Prm Data
3.21.	(m)	Attribute:	Cfg Data Len
3.22.	(m)	Attribute:	Cfg Data
3.23.	(m)	Attribute:	Add Tab Len
3.24.	(m)	Attribute:	Add Tab
3.25.	(m)	Attribute:	Slave User Data Len
3.26.	(o)	Attribute:	Slave User Data
3.27.	(o)	Attribute:	Ext User Prm Data Len
3.28.	(o)	Attribute:	Ext User Prm Data

3.29.	(m)	Attribute:	List of assigned CREPs for the MS0 AR
3.29.1.	(m)	Attribute:	CREP
3.30.	(m)	Attribute:	Assigned AREP for the MS1 AR
3.31.	(m)	Attribute:	Assigned AREP for the MS3 AR
4.	(m)	Attribute:	Assigned AREP for the MS0 AR

SERVICES:

1.	(o)	OpsService:	Start Seq
2.	(o)	OpsService:	Download
3.	(o)	OpsService:	Upload
4.	(o)	OpsService:	End Seq
5.	(o)	OpsService:	Act Para Brct
6.	(o)	OpsService:	Act Param

6.2.6.2.2.3 Attributes

Implicit
This attribute indicates that the Master Parameter object is implicitly addressed by the services.

Bus Para
This combined attribute consists of the following items.

Bus Para Len
This attribute contains the length of Bus Para inclusive the length parameter.

Attribute Type: Unsigned16

Allowed values: 66 to $2^{16} - 1$

TS
This attribute contains the own DL-address of the DP-master.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Data rate
This attribute contains the code number of the chosen data rate.

Attribute Type: Unsigned8

The allowed values are shown in Table 136.

Table 136 – Data rate

Value	Meaning
0	9,6 kbit/s
1	19,2 kbit/s
2	93,75 kbit/s
3	187,5 kbit/s
4	500 kbit/s
6	1 500 kbit/s
7	3 000 kbit/s
8	6 000 kbit/s
9	12 000 kbit/s
10	31,25 kbit/s
11	45,45 kbit/s

T_{SL}, min T_{SDR}, max T_{SDR}, T_{QUI}, T_{SET}, T_{TR}, G, HSA, max retry limit
These attributes are described in IEC 61158-3-3 and IEC 61158-4-3.

Error Action Flag
This attribute reflects the reaction of the DP-master (Class 1) in case of failed data transfers. This attribute will be evaluated in the operation mode OPERATE and CLEAR of the DP-master (Class 1). If the data transfer to at least one activated DP-slave was impossible for a certain period of time (Data Control Time) and if the Error Action Flag is set, the operation mode will change from OPERATE to CLEAR. If the Error Action Flag is cleared, the DP-master (Class 1) will remain in the operation mode OPERATE despite erroneous data transfers.

If the operation mode of the DP-master (Class 1) shall be changed from CLEAR to OPERATE with the Error Action Flag set to TRUE the data transfer to all activated DP-slaves have to be performed successfully for a certain period of time (Data Control Time)

The allowed values are shown in Table 137.

Table 137 – USIF state

Value	Meaning
0	no change of the operation mode in case of an error
1	change of the operation mode in case of an error

Isochronous Mode
This attribute indicates if the DP-master (Class 1) shall operate in the Isochronous Mode. Two different models concerning the interface between the DP Application Layer and the application of the DP-master (Class 1) itself are provided by fieldbus DP for the DP-master (Class 1), see 6.2.2.4.3.2.

The allowed values are shown in Table 138.

Table 138 – Isochronous mode

Value	Meaning
Not Synchronized	the DP-master (Class 1) shall not operate in the Isochronous Mode
Buffered Synchronized	the DP-master (Class 1) shall operate in the Buffer Synchronized Isochronous Mode
Enhanced Synchronized	the DP-master (Class 1) shall operate in the Enhanced Synchronized Isochronous Mode

IsoM Freeze
This Boolean attribute indicates if the synchronization message (SYNCH) shall be sent with the Control Command Freeze set to TRUE.

Allowed values: TRUE, FALSE

IsoM Sync
This Boolean attribute indicates if the synchronization message (SYNCH) shall be sent with the Control Command Sync set to TRUE.

Allowed values: TRUE, FALSE

Min Slave Interval
This attribute specifies the smallest allowed period of time between two consecutive DP-slave poll cycles. This ensures that the sequence of cyclic service requests from the DP-master (Class 1) can be handled by the DP-slave. This period of time will be complied by the DP-master (Class 1) for every cyclic Master-slave service except for the Global Control service. For the Global Control service the User is responsible for compliance with Min Slave Interval.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 100 µs

Poll Timeout
In the case of MM1 AR this attribute specifies the maximum period of time needed to get the response.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 1ms

Data Control Time
This attribute specifies the maximum period of time for an Data Exchange with every activated DP-slave. This Time is used for sending out the Global Control cyclically.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 10 ms

Alarm Max
This variable contains the maximum number of alarms per DP-slave, which can be handled by the DP-master. It describes a Master capability. The Master shall be able to handle a minimum number of 7 alarms, because there are existing 7 different alarm classes.

Attribute Type: Unsigned8

Allowed values: 7 to 32

Max User Global Control
This attribute defines the maximum number of Global Control requests, which may be started by the User at the same time. This parameter describes the ability of the DP-master (Class 1). A practicable value for Max User Global Control is 16. For each of the 8 Slave groups one Sync and one Freeze Command may be started at the same time.

Attribute Type: Unsigned8

Allowed values: 1 to 255

Master User Data Len
This attribute contains the length of Master User Data inclusive the length parameter.

Attribute Type: Unsigned16

Allowed values: 34 to 2^{16} - 1

Master Class2 Name
This attribute indicates the name of the DP-master (Class 2) which provided the values for the attributes of this Master Parameter object.

Attribute Type: Visible String(32)

Master User Data
This attribute contains manufacturer specific data which is necessary for the bus parameter set.

Attribute Type: Octet String

T_{CT}
This attribute specifies the time (in units of t_{BIT}) that is necessary to handle a complete Isochronous DP cycle and corresponds to the time T_{DP}.

Allowed values: 1 to 2^{24} - 1

maxT_{SH}
This attribute specifies the maximum allowed time shift (in units of t_{BIT}) of the time T_{DP} measured by the DP-master (Class 1). The application of the DP-master (Class 1) will be informed if the measured Isochronous DP cycle time exceeds this limit by the service SYNCH Delayed indicating the occurred shift.

Allowed values: 1 to 2^{8} - 1

Slave Para
This combined attribute consists of the following items.

Slave DL-address
This attribute reflects the DL-address of the DP-slave. This address is used as an identifier (Area Code) to address the corresponding slave parameter set, e.g. for up- or download purposes.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Slave Para Len
This attribute contains the length of Slave Para inclusive the length parameter. A slave parameter set can be deleted by setting the Slave Para Len to zero.

Attribute Type: Unsigned16

Allowed values: 0 to 2^{16} - 1

Extra Alarm DLSAP
This Boolean attribute selects the Channel for the Alarm Ack service.

- If the attribute is set to FALSE the DP-master (Class 1) acknowledges alarms via DLSAP 51.

If the attribute is set to TRUE the DP-master (Class 1) acknowledges alarms via DLSAP 50.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

DPV1 Data Types
This Boolean attribute reflects whether the DP-slave uses data types.

- If the attribute is set to FALSE the DP-slave uses only basic types (Octet, Word, Array) for the Data Description.

If the attribute is set to TRUE the DP-slave uses data types for the Data Description.

Attribute Type: Boolean

DPV1 Supported
This Boolean attribute reflects whether the DP-slave uses the extended functionality.

- If the attribute is set to FALSE the DP-slave uses no extended functionality.

If the attribute is set to TRUE the DP-slave uses extended functionality.

Attribute Type: Boolean

Fail Safe
This Boolean attribute reflects whether the DP-slave supports the Fail Safe functionality. If the attribute is set to FALSE the DP-slave does not support the Fail Safe functionality. If the attribute is set to TRUE the DP-slave supports the Fail Safe functionality.

Attribute Type: Boolean

Active
This Boolean attribute reflects whether the DP-slave has to be activated. If the attribute Active has the value FALSE, the DP-master (Class 1) stops the DP-slave handling to this assigned DP-slave. In this state it is allowed to change the slave parameter set, for example by a DP-master (Class 2) with the Download service. If the Active-Flag has the value TRUE, the DP-slave handling will be done by the DP-master (Class 1) to this assigned DP-slave. Changing of the slave parameter set in this state is limited (see also the description of the attribute New Prm).

Attribute Type: Boolean

Slave Type
This attribute contains a manufacturer specific type denotation for the DP-slave see Table 139.

Attribute Type: Unsigned8

The allowed values are shown in Table 139.

Table 139 – Slave type

Value	Meaning
0	DP-slave
1-15	reserved
16-255	manufacturer specific

New Prm
This Boolean attribute reflects whether parameterization data has to be transferred to the DP-slave in the data exchange mode. If this attribute is set to FALSE the Output Data will be transferred to this DP-slave in the data exchange mode. If this attribute is set to TRUE, for one data cycle parameterization data will be transferred instead of Output Data. The consistency of the parameterization data has to be guaranteed by local measures. After transmitting this parameterization data successfully the DP-master (Class 1) sets this attribute to FALSE and the data exchange mode to the DP-slave will be continued.

Attribute Type: Boolean

Max Diag Data Len
This attribute contains the maximum length of Diag Data for the DP-slave.

Attribute Type: Unsigned8

Allowed values: 6 to 244

Max Channel Data Length
This attribute defines the maximum length of a MS1-APDU (Length parameter and 4 Octet Header) for the corresponding DP-slave.

Attribute Type: Unsigned8

Allowed values: 4 to 244

Diag Upd Delay
The attribute reflects the number of Diag service requests performed in the state DIAG2 of the DP-master (Class 1) while Prm Req is set in Diag Data of the DP-slave (for DP-slaves with reduced performance).

Attribute Type: Unsigned8

Allowed values: 0 to 15 (extendible up to 255)

Alarm Mode
This attribute specifies the maximum number of possible active alarms.

Attribute Type: Unsigned8

The allowed values are shown in Table 140.

Table 140 – Alarm mode

Value	Meaning
0	1 alarm of each type
1	2 alarms in total
2	4 alarms in total
3	8 alarms in total
4	12 alarms in total
5	16 alarms in total
6	24 alarms in total
7	32 alarms in total

NA To Abort
This Boolean attribute reflects the behavior of the DP-master (Class 1) if the assigned DP-slave does not respond at the MS0 AR in the DIAG2 state and the data exchange mode.

- If this attribute is set to FALSE the DP-master continues in case of no response (Status = NA) of the assigned DP-slave.

If this attribute is set to TRUE the DP-master aborts the MS0 AR in case of no response (Status = NA) of assigned the DP-slave (abort means sending an unlock command and to re-establish the MS0 AR).

Attribute Type: Boolean

Ignore AClr
This Boolean attribute in combination with the Error Action Flag allows a DP-slave specific reaction of the DP-master (Class 1) in case of failed data transfers.

- If the attribute is set to FALSE the DP-master (Class 1) processes the auto-clear function as defined in the Error Action Flag.

If the attribute is set to TRUE the DP-master (Class 1) ignores the auto-clear function as defined in the Error Action Flag.

Attribute Type: Boolean

Publisher Flag
This Boolean attribute indicates if the correspondent DP-slave shall operate as Publisher. If the attribute is set to FALSE the correspondent DP-slave shall not operate as Publisher. If the attribute is set to TRUE the correspondent DP-slave shall be addressed as Publisher by the DP-master (Class 1).

Attribute Type: Boolean

Allowed values: TRUE, FALSE

MS1 Timeout
This attribute contains the value for the Timeout of the acyclic communication on the MS1 AR and specifies the maximum period of time it may take a DP-slave to provide a response. The time contained in this attribute is the maximum time of the assigned DP-slave.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 10ms

Prm Command supported
This Boolean attribute reflects whether the DP-slave uses PrmCmd within the extended functionality:

- If the attribute is set to FALSE the DP-slave supports no PrmCmd.

If the attribute is set to TRUE the DP-slave supports PrmCmd.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Prm Data Len
This attribute contains the length of Prm Data inclusive the length parameter.

Attribute Type: Unsigned8

Allowed values: 9 to 246

Prm Data
This attribute reflects the parameterization data sent to the corresponding DP-slave by the DP-master (Class 1). The structure of this attribute is described in 6.2.10.4.2.2.

Cfg Data Len
This attribute contains the length of Cfg Data inclusive the length parameter.

Attribute Type: Unsigned8

Allowed values: 3 to 246

Cfg Data
This attribute reflects the configuration information sent to the DP-slave. The structure of this attribute is described in 6.2.10.4.2.2.

Add Tab Len
This attribute contains the length of Add Tab inclusive the length parameter.

Attribute Type: Unsigned16

Allowed values: 2 to 2^{16} - 31

Add Tab
This attribute contains the address assignment table of the DP-slave. In case of a Programmable Logic Controller, this list contains the Programmable Logic Controller addresses of the decentralized addresses.

Slave User Data Len
This attribute contains the length of the Slave User Data inclusive the length parameter.

Attribute Type: Unsigned16

Allowed values: 2 to 2^{16} - 31

Slave User Data
This attribute contains manufacturer specific data that characterize the DP-slave for the DP-master (Class 1).

Attribute Type: Octet String

Ext User Prm Data Len
This attribute contains the length of Ext Prm Data inclusive the length parameter.

Attribute Type: Unsigned8

Allowed values: 2 to 246

Ext User Prm Data
This attribute reflects the extended user parameterization data sent to the correspondent DP-slave by the DP-master (Class 1). The DP-master (Class 1) will send this data between the parameter data and the configuration information.

List of assigned CREPs for the MS0 AR
This attribute contains the CREPs of the MS0 AR assigned for this DP-slave and is composed of the following list elements:

CREP
This attribute contains one CREP of the MS0 AR.

Assigned AREP for the MS1 AR
This attribute contains the AREP of the MS1 AR assigned for this DP-slave.

Assigned AREP for the MS3 AR
This attribute contains the AREP of the MS3 AR assigned for this DP-slave.

Assigned AREP for the MS0 AR
This attribute contains the AREP of the MS0 AR assigned for all DP-slaves.

6.2.6.2.2.4 Invocation of the master parameter object

For the invocation of the Master Parameter object the following rules apply:

— Only one Master Parameter object can be invoked in a DP-master (Class 1).

— Up to 125 Slave Para elements with an unique value for the attribute Slave DL-address can be contained in a Master Parameter object of a DP-master (Class 1).

6.2.6.3 Management service specification

6.2.6.3.1 Get master diag

With this service a DP-master (Class 2) can read a single element (Master Status, System Diagnosis, Data Transfer List, DP-slave Diag of a single DP-slave) from the Master Diag object of a DP-master (Class 1). This service shall only be used in conjunction with the MM1 AR.

Table 141 shows the parameters of the service.

Table 141 – Get Master Diag

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
MDiag Identifier	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Diagnosis Data			M	M(=)
DP-slave Diag			S	S(=)
System Diagnosis			S	S(=)
Data Transfer List			S	S(=)
Master Status			S	S(=)
Result(-)			S	S(=)
AREP			M	M(=)
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

MDiag Identifier
This parameter specifies the Element of the Master Diag object to be read.

The allowed values are shown in Table 142.

Table 142 – MDiag identifier

Value	Meaning
0-125	DP-slave Diag of the DP-slave with the corresponding DL-address
126	System Diagnosis
127	Master Status
128	Data Transfer List

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

Diagnosis Data
This parameter contains one of the following element of the Master Diag object as requested.

DP-slave Diag
This parameter contains the Diag Data of the addressed DP-slave.

System Diagnosis
This parameter contains the System Diagnosis of the Master Diag object (see 6.2.6.2.1)

Master Status
This parameter contains the Master Status of the Master Diag object (see 6.2.6.2.1)

Data Transfer List
This parameter contains the Data Transfer List of the Master Diag (see 6.2.6.2.1)

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS, NA, RS, RR, UE, RE, TO, FE, NE, IP, AD, EA, LE

6.2.6.3.2 Start seq

With the Start Seq service the beginning of an Upload or Download sequence is signalized by the DP-master (Class 2) in the DP-master (Class 1). In this case the DP-master (Class 1) can set an access protection for the addressed area until the End Seq service signals the end of the sequence. The addressed data can be transmitted in blocks with the services Upload and Download. This service shall only be used in conjunction with the MM1 AR.

Table 143 shows the parameters of the service.

Table 143 – Start Seq

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M		
AREP	M	M		
Area Code	M	M(=)		
Timeout	M	M(=)		
Result(+)			S	S(=)
AREP			M	M
Max Len Data Unit			C	C
Result(-)			S	S(=)
AREP			M	M
Status			M	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

Area Code
The parameter Area Code identifies the element of the Master Parameter object which shall be loaded or read. If the parameter Area Code is set to a value of 255, local access protection for a determined Area Code is not guaranteed for the services following. The local access protection ensures that the User of the DP-master (Class 1) cannot access the partially loaded data during the service sequence.

Attribute Type: Unsigned8

The allowed values are shown in Table 144.

Table 144 – Area code (start seq)

Value	Meaning
0 to 125	Slave Para of the DP-slave with the corresponding DL-address
126	reserved
127	Bus Para
128	reserved
129	statistic counters
130 to 135	reserved
136 to 139	for the transfer of the Master Parameter Set
140 to 254	reserved
255	Start Seq: No local access protection

Timeout
This parameter defines the control time between two successive Upload/Download services. If this time expires the access protection is deactivated. The previously transferred data are invalid.

Attribute Type: Unsigned16

Time Base: 1ms

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

Max Len Data Unit
The parameter Max Len Data Unit defines the maximum possible length of the parameter Data in the following Upload/Download APDUs.

Attribute Type: Unsigned8

Range: 1 to 240

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS, NA, RS, RR, UE, TO, FE, RE, NE, AD, IP, NI, SE, SC, EA, LE

6.2.6.3.3 Download

This service is used to transfer the values for the attributes of the Master Parameter object from the DP-master (Class 2) to a DP-master (Class 1). The beginning of the Download sequence can be signaled with the service Start Seq. The actual data set is transmitted in blocks with the Download service in the addressed area. After the transfer is completed the sequence shall be terminated with the End Seq service, if the Download sequence was started with the Start Seq service. This service shall only be used in conjunction with a MM1 AR.

Table 145 shows the parameters of the service.

Table 145 – Download

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M		
AREP	M	M		
Area Code	M	M		
Add Offset	M	M		
Data	M	M		
Result(+)			S	S(=)
AREP			M	M
Result(-)			S	S(=)
AREP			M	M
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

Area Code
The parameter Area Code identifies the element of the Master Parameter object which shall be loaded.

Attribute Type: Unsigned8

Allowed values: see 6.2.6.3.2

Add Offset
This parameter contains the offset to the begin of the area which is specified by the Area Code.

Attribute Type: Unsigned16

Data
This parameter contains the data to be transferred.

Attribute Type: Octet String

Length: 1 to 240

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

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS, NA, RS, RR, UE, TO, FE, RE, NE, AD, EA, LE, SC, NI, NC

6.2.6.3.4 Upload

This service is used to transfer the values of the attributes of the Master Parameter object from the DP-master (Class 1) to a DP-master (Class 2). The Upload sequence can be

executed in the same way as the Download sequence. It is possible to transfer single data areas without the services Start Seq and End Seq. In this case the consistency of the transferred data is not guaranteed. This service shall only be used in conjunction with a MM1 AR.

Table 146 shows the parameters of the service.

Table 146 – Upload

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M		
AREP	M	M		
Area Code	M	M		
Add Offset	M	M		
Data Len	M	M		
Result(+)			S	S(=)
AREP			M	M
Data			C	C(=)
Result(-)			S	S(=)
AREP			M	M
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

Area Code
The parameter Area Code identifies the element of the Master Parameter object which shall be read

Attribute Type: Unsigned8

Allowed values: see 6.2.6.3.2

Add Offset
This parameter identifies the offset to the begin of the area which is specified by the Area Code.

Attribute Type: Unsigned16

Data Len
The parameter defines the length of the requested data.

Attribute Type: Unsigned8

Allowed values: 1 to 240

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

Data
This parameter contains the requested data.

Attribute Type: Octet String

Length: 1 to 240

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS, NA, RS, RR, UE, TO, FE, RE, NE, EA, LE, NI, SC, AD

6.2.6.3.5 End seq

This service is used to mark the end of a sequence of consistent transfers of data blocks of the Master Parameter object. This service shall only be used in conjunction with a MM1 AR.

Table 147 shows the parameters of the service.

Table 147 – End Seq

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M		
AREP	M	M		
Result(+)			S	S(=)
AREP			M	M
Result(-)			S	S(=)
AREP			M	M
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

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

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS, NA, RS, RR, UE, TO, FE, RE, NI, SE, NE, AD, EA, LE, NC

6.2.6.3.6 Act Para Brct

This service is used to activate the previously transferred values for the Master Parameter object by the DP-master (Class 2). This service may be performed simultaneously to one (single) or many (multicast) DP-master (Class 1). In the operation modes CLEAR and OPERATE of the DP-master (Class 1), the activation of the Bus Para element with changed DL parameters data rate or TS will be rejected. This service is only locally confirmed because a possible change of the data rate shall take effect at all addressed DP-masters (Class 1) at the same time. This service shall only be used in conjunction with a MM2 AR.

Table 148 shows the parameters of the service.

Table 148 – Act Para Brct

Parameter name	Req	Ind	Cnf
Argument	M	M	
AREP	M	M	
Area Code	M	M	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

Area Code
The parameter Area Code identifies the element of the Master Parameter object which shall be activated.

Attribute Type: Unsigned8

The allowed values are shown in Table 149.

Table 149 – Area code (Act Para Brct)

Value	Meaning
0 to 126	not allowed
127	Bus Para
128 to 129	reserved
130 to 135	reserved
136 to 139	for the activation of the Master parameter object
140 to 254	reserved
255	not allowed

In the presence of several DP-masters (Class 2) it shall be guaranteed that the old and the new bus parameter sets are compatible.

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

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS

6.2.6.3.7 Act param

This service is used by the DP-master (Class 2) for the following purposes:

— to activate or to deactivate a parameterized DP-slave,

— to change the operation mode of the DP Application Layer,

— to accept and activate the values for the Bus Para Elements of the Master Parameter object.

If the loaded bus parameter set contains a changed data rate or station address for the DP-master (Class 1) the new bus parameter set shall be activated with the Act Para Brct service. This service shall only be used in conjunction with a MM1 AR.

Table 150 shows the parameters of the service.

Table 150 – Act param

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M		
AREP	M	M		
Area Code	M	M(=)		
Activate	M	M(=)		
Result(+)			S	S(=)
AREP			M	M
Result(-)			S	S(=)
AREP			M	M
Status			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier of the desired AR.

Area Code
The parameter Area Code identifies the element of the Master Parameter object which shall be activated/deactivated with the exception of the operation mode.

Attribut Type: Unsigned8

The allowed values are shown in Table 151.

Table 151 – Area code (Act param)

Value	Meaning
0 to 125	Slave Para of the DP-slave with the corresponding DL-address The (Flag/Bit) Active in the slave parameter set of the DP-master (Class 1) is set as specified in the parameter Activate.
126	Reserved
127	Bus Para
128	Operation mode (description see 6.2.5.3.29)
129	Reserved
130 to 135	reserved
136 to 139	activation of the Master Parameter Set
140 to 255	reserved

Activate
The meaning and the allowed values of this parameter are depending on the Area Code.

Attribute Type: Unsigned8

The allowed values are shown in Table 152.

Table 152 – Activate

Area Code	Allowed values for the parameter Activate	
0 to 125: (Slave Para)	0	(= Deactivate)
	128	(= Activate)
127: (Bus Para)	255	(= Activate)
128: (operation mode)	64	(= STOP)
	128	(= CLEAR)
	192	(= OPERATE)

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

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

Status
This parameter provide detailed information about the failure.

Allowed values: DS, NA, RS, RR, UE, TO, FE, RE, NE, AD, IP, SC, NI, DI, EA, LE

6.2.7 Load region ASE

6.2.7.1 Overview

A Load Region object represents an unstructured memory area whose contents may be uploaded (Pull) or downloaded (Push). Unstructured in this context means that the memory area is represented only as an ordered sequence of octets. No other structure is apparent.

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

The Load Region objects shall clearly be separated from the Process Data objects and the Function Invocation objects. No overlapping is allowed.

To maintain integrity, only one download process for a Load Region object is permitted at a time. The Load Region State attribute of the Load Region object is used to indicate whether the Load Region object is empty, being downloaded, or loaded (see the definition of the attribute for a complete list of the states). The Load Region object may be cleared using the Initiate Load service.

The Load Region State attribute also indicates the progress of an upload. Multiple uploads of a Load Region object via different ARs are allowed. This attribute is defined to separate the state of the contents of the Load Region object from the operation of having the contents uploaded.

The Load Region ASE provides services that permit an AP to initiate the upload or download (Pull or Push) of Load Region objects. They identify the Load Region object and whether to Pull or Push as parameters of the initiate load request.

The LR Data may be transferred in segments by using the Pull Segment or the Push Segment service.

LR Data segments are uploaded by the Client issuing Pull Segment requests for them. The AP that contains the LR Data responds by returning the requested segment, indicating with a parameter when the final segment of the LR Data has been returned.

LR Data segments are downloaded by the Client having the AP containing the LR Data image by sending individual segments to the Server and wait for a response that indicates whether the LR Data segment was received. In this case, the Client indicates when the last LR Data segment has been transferred. After the last LR Data segment has been received, the AP that initiated the transfer ends the load process by issuing a Terminate Load request to the remote AP. The Terminate Load request may also be used by an AP if the load process cannot be completed successfully.

To avoid deadlocks at the Server a timeout shall be issued to control the activities of an AR regarding the upload or download procedures.

6.2.7.2 Load region class specification

6.2.7.2.1 Template

The Load Region object is modeled as a Server object. The Load Region object specifies the attributes and services used by Client APs to upload and/or download LR Data. The specification of the Client AP is beyond the scope of this specification.

A Load Region object is described by the following template:

DP ASE:			**Load Region ASE**
CLASS:		Load Region	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Identifier
2	(m)	Attribute:	LR Size
3.	(m)	Attribute:	Access Rights MS1
4.	(m)	Attribute:	Access Rights MS2
5.	(c)	Constraint:	Access Rights MS2 = Right to Pull or Right to Pull/Push
5.1.	(m)	Attribute:	List of SCL Pull
5.1.1.	(m)	Attribute:	SCL
6.	(c)	Constraint:	Access Rights MS2 = Right to Push or Right to Pull/Push
6.1.	(m)	Attribute:	List of SCL Push

6.1.1.	(m)	Attribute:	SCL
7.	(m)	Attribute:	Load Region State
8.	(m)	Attribute:	Actual LR Size
9.	(o)	Attribute:	Identifier of Related Function Invocation
10.	(m)	Attribute:	Max Segment Length
11.	(m)	Attribute:	Max Response Delay
12.	(m)	Attribute:	Uploadcount
13.	(o)	Attribute:	Load Region Additional Information
14.	(o)	Attribute:	Load Region Name
15.	(o)	Attribute:	Local Detail

SERVICES:

1.	(m)	OpsService:	Initiate Load
2.	(o)	OpsService:	Push Segment
3.	(o)	OpsService:	Pull Segment
4.	(m)	OpsService:	Terminate Load

6.2.7.2.2 Attributes

Identifier
This key attribute is a triplet composed of API, Slot Number and LR Index to define to which Application Process Identifier, module and to which index within the specified module the Load Region object belongs. This Identifier shall not be used by another Action object, Load Region object, Function Invocation object or Process Data object.

Attribute Type: Unsigned8, Unsigned8, Unsigned16

Allowed triplet values: [0 to 254, 0 to 254, 0 to 65 199 (65 200 to 65 535 reserved)]

LR Size
This attribute specifies the length of the Load Region object in octets.

Attribute Type: Unsigned32

Allowed values: 1 to $2^{32} - 1$

Access Rights MS1
This attribute defines the type of access rights defined for a MS1 AR for the Load Region object.

Attribute Type: Unsigned8

The allowed values are shown in Table 153.

Table 153 – Access rights MS1

Value	Meaning
0	Right to Push
1	Right to Pull
2	Right to Pull/Push
3	No Access

Access Rights MS2
This attribute defines the type of access rights defined for a MS2 AR for the Load Region object.

Attribute Type: Unsigned8

The allowed values are shown in Table 154.

Table 154 – Access rights MS2

Value	Meaning
0	Right to Push
1	Right to Pull
2	Right to Pull/Push
3	No Access

List of SCL Pull
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one value for Pull access protection for this Load Region object. The rules for accepting or rejecting a Pull access are described in 6.2.1.3 according to the read access.

Attribute Type: Unsigned8

List of SCL Push
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one value for Push access protection for this Load Region object. The rules for accepting or rejecting a Push access are described in 6.2.1.3 according to the write access protection.

Attribute Type: Unsigned8

Load Region State
This attribute specifies the state of the Load Region object.

Attribute Type: Unsigned8

The allowed values of this attribute are shown in Table 155.

Table 155 – Load region state

Value	Meaning
0	DOWNLOADABLE
1	DOWNLOADING
2	DOWNLOAD SUCCESS
3	LOADED
4	IN-USE

Load Region State attribute constraints:

- A Load Region object whose contents are stored in non-erasable memory cannot support downloading transfers. Its valid state values are LOADED and IN USE.
- A Load Region object whose contents are stored in erasable memory supports downloading transfers. Its valid state values are DOWNLOADABLE, DOWNLOADING, DOWNLOADING SUCCESS, LOADED and IN USE,

Actual LR Size
This attribute specifies the used length of the Load Region object in octets. A Load Region object whose content is cleared shall have a value of zero for this attribute.

Attribute Type: Unsigned32

Allowed values: 0 to 2^{32} - 1

Identifier of Related Function Invocation
This optional attribute is a triplet composed of API, Slot Number and FI Index to define to which Application Process Identifier, module and to which index within the specified module the related Function Invocation object belongs.

Attribute Type: Unsigned8, Unsigned8, Unsigned16

Allowed triplet values: Allowed values:[0 to 254, 0 to 254, 0 to 65 199 (65 200 to 65 535 reserved)]

Max Segment Length
This attribute specifies the maximum length in octets of each Push Segment or Pull Segment which shall be used for the up- or download of this Load Region object.

Attribute Type: Unsigned8

Allowed values: 1 to 234

Max Response Delay
This attribute specifies the maximum time period (in units of 10ms) the response of an Initiate Load, Pull Segment, Push Segment or Terminate Load service request may be delayed by the Server due to processing of the request. The value of this attribute shall be less than the value of the attribute MS1 Timeout of the ARL DP-master (Class 1) object.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Uploadcount
This attribute indicates the number of Clients which currently upload the contents of the Load Region object. The value of this attribute has to be zero to download the Load Region object. This attribute is incremented each time a Client uploads the Load Region object in parallel. This attribute is decremented each time a Client finishes the upload of the Load Region object.

Attribute Type: Unsigned8

Allowed values: 0 to 255

Load Region Additional Information
This optional attribute contains additional information about the Load Region object, e.g. permitted data formats.

Attribute Type: Octet String

Load Region Name
This optional attribute defines the name of the Load Region object.

Attribute Type: Visible String(32)

Local Detail
This optional attribute contains additional local information about the Load Region object.

Attribute Type: Octet String

6.2.7.2.3 Invocation of the load region object

For the invocation of the Load Region object the following rules apply.

— Load Region objects shall not exceed the total length of 2^{32} - 1 octets. The attribute LR Size has to be set accordingly.

— At least the Push Segment or Pull Segment or both services shall be allowed for the access of the Load Region object.

— At least one AR shall have access to the Load Region object.

— Only one Function Invocation shall be related to a Load Region object.

— The access rights shall be set accordingly to the allowed service(s).

6.2.7.3 Load region service specification

6.2.7.3.1 Initiate load

This service is used by an AP to initiate the download or upload of a Load Region. A parameter of the service indicates whether the load image will be transferred into the Load Region object using the Pull Segment service or from the Load Region object using the Push Segment service. This service shall only be used in conjunction with the MS1 or MS2 AR. Table 156 shows the parameter of the service.

Table 156 – Initiate load

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
LR Index	M	M(=)		
Load Type	M	M(=)		
Load Image Size	C	C(=)		
Intersegment Request Timeout	M	M(=)		
Additional Information	U	U(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Actual LR Size			M	M(=)
Max Response Delay			M	M(=)
Max Segment Length			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Load Region object in a specific slot (typically a module).

LR Index
This parameter specifies the index of the Load Region to be downloaded or to be uploaded.

Load Type
This parameter specifies whether the Load Region object has to be downloaded or uploaded.

Allowed values: Push, Pull

Load Image Size
This conditional parameter indicates the number of octets which has to be downloaded to the Load Region object if the parameter Load Type is equal Push. If the parameter Load Type is equal Pull the parameter Load Image Size is not present. This parameter shall have a value of zero if the content of the Load Region object shall be cleared.

Attribute Type: Unsigned32

Allowed values: 0 to 2^{32} - 1

Intersegment Request Timeout
This parameter specifies the timeout period on the Server side between Pull Segment or Push Segment service requests, in units of 100ms. The default values are shown in Table 157.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Table 157 – Default values for the parameter Intersegment Request Timeout

Baudrate (kbit/s)	≤187,5	500	1 500	3 000	6 000	12 000
Intersegment Request Timeout (sec)	40	20	10	10	10	10

Additional Information
This optional parameter specifies additional information for the subsequent load procedure, e.g. the desired data format.

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

Actual LR Size
This parameter contains the value

— of the attribute Actual LR Size of the Load Region object if the parameter Load Type has had the value Pull in the previous request primitive

— of the parameter Load Image Size if the parameter Load Type has had the value Push in the previous request primitive.

Attribute Type: Unsigned32

Allowed values: 0 to 2^{32} - 1

Max Response Delay
This parameter contains the value of the attribute Max Response Delay of the Load Region object. The Client AP shall monitor the Initiate Load, Push Segment, Pull Segment or Terminate Load service request and response sequences with this time parameter.

Max Segment Length
This parameter contains the value of the attribute Max Segment Length of the Load Region object.

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: state conflict, load image size error, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type.

6.2.7.3.2 Push segment

This optional service is used to transfer LR Data to the Load Region object after the download process has been initiated. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 158 shows the parameter of the service.

Table 158 – Push segment

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
LR Index	M	M(=)		
Segment Length	M	M(=)		
Segment Number	M	M(=)		
More Follows	M	M(=)		
Data	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Load Region object in a specific slot (typically a module).

LR Index
This parameter specifies the index of the Load Region object to be downloaded.

Segment Length
This parameter indicates the number of octets of the Load Region object which has to be downloaded.

Attribute Type: Unsigned8

Allowed values: 1 to 234

Segment Number
This parameter indicates the number of the segment which is being transferred. Segment numbers are in ascending order without gaps beginning with the value one.

Attribute Type: Unsigned32

Allowed values: 1 to 2^{32} - 1

More Follows
This parameter indicates whether or not any additional segments remain to be transmitted.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Data
The parameter Data contains the segment of the Load Image which has to be downloaded and consists of the number of octets indicated in the parameter Segment Length of the request. This parameter has to be composed of the data type Octet String.

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

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: state conflict, length error, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type, feature not supported.

6.2.7.3.3 Pull segment

This optional service is used to transfer LR Data from the Load Region object after the upload process has been initiated. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 159 shows the parameter of the service.

Table 159 – Pull segment

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
LR Index	M	M(=)		
Segment Length	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Segment Length			M	M(=)
Segment Number			M	M(=)
More Follows			M	M(=)
Data			C	C(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Load Region object in a specific slot (typically a module).

LR Index
This parameter specifies the index of the Load Region to be uploaded.

Segment Length
This parameter indicates the number of octets of the Load Region object which shall be uploaded with this request.

Attribute Type: Unsigned8

Allowed values: 1 to 234

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

Segment Length
The value of this parameter is equal to the equivalent parameter of the request if the remaining octets to be uploaded are greater or equal to the requested segment length. Otherwise this parameter contains the number of remaining octets.

Attribute Type: Unsigned8

Allowed values: 0 to 234

Segment Number
This parameter indicates the number of the segment which is being transferred. Segment numbers are in ascending order without gaps beginning with the value one.

Attribute Type: Unsigned32

Allowed values: 1 to 2^{32} - 1

More Follows
This Boolean parameter indicates whether or not any additional segments remain to be transmitted.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Data
This conditional parameter is only present if the value of the parameter Segment Length of the response is different from zero and contains the segment of the Load Image which has to be uploaded and consists of the number of octets indicated in the parameter Segment Length of the response. This parameter has to be composed of the data type Octet String.

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

Error Code
This parameter indicates the reason for the failure.

Allowed value: state conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type, feature not supported.

6.2.7.3.4 Terminate load

This service is used to terminate the upload or download process. It shall be used upon successful completion, or to abort an upload or download in progress.

If the up- or download process is being aborted with the Terminate Load service, then

— in case of a download the contents of the Load Region object shall be deleted (the attribute Actual LR Size shall be set to zero and the attribute Load Region State shall be set to DOWNLOADABLE);

— in case of an upload the value of the attribute Uploadcount of the Load Region object shall be decremented.

This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 160 shows the parameter of the service.

Table 160 – Terminate load

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
LR Index	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Load Region object in a specific slot (typically a module).

LR Index
This parameter specifies the index of the Load Region object whose upload or download process shall be terminated.

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

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

Error Code
This parameter indicates the reason for the failure.

Allowed value: state conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type.

6.2.7.4 Behavior of the load region object

6.2.7.4.1 Load region state machine description

6.2.7.4.1.1 Overview

The behavior of a Load Region object is defined by the Load Region state machine. Each Load Region object is represented by an own instance of this state machine.

The Load Region state machine represents a part of the User functionality and is included to ensure interoperability. It is assumed that other parts of the User functionality perform the following checks:

— validation of the existence and addressing of the Load Region object

— validation of the access rights

— validation of temporary system restrictions for the use of the Load Region object

and generate appropriate Error Codes if necessary.

Two types of Load Region state machines are defined. One represents the behavior of Load Region objects whose contents are stored in erasable memory and are therefore capable to be downloaded. The other one represents the behavior of Load Region objects whose contents are stored in non erasable memory and are therefore not capable to be downloaded.

The Load Region object may be used by Function Invocation objects. Therefore the Load Region state machines provide an interface to the Function Invocation state machines for co-ordination purposes.

The main functionalities of the Load Region state machine are:

— monitoring of the valid sequence of services executed on the Load Region object
— to ensure that a download is only executed if no upload is in progress
— to ensure that an upload or download is only executed if the Load Region object is not used by a Function Invocation object
— to ensure that an upload is only executed if no download is in progress
— supervision of the Client activity.

6.2.7.4.1.2 Primitive definitions

6.2.7.4.1.2.1 Primitives exchanged between the user and the load region state machine

Table 161 and Table 162 show the primitives exchanged between the User and the Load Region state machine.

Table 161 – Primitives issued by the user to the Load Region state machine

Primitive Name	Source	Associated parameters	Functions
Abort.ind	User	AREP Locally Generated Subnet Instance Reason Code Additional Detail	—
DP-slave Stopped.ind	User	AREP	—
Initiate Load.ind	User	AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request, Timeout	—
Pull Segment.ind	User	AREP, Slot Number, LR Index, Segment Length	—
Push Segment.ind	User	AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data	—
Terminate Load.ind	User	AREP, Slot Number, LR Index	—

Table 162 – Primitives issued by the Load Region state machine to the user

Primitive Name	Source	Associated parameters	Functions
Initiate Load.rsp(+)	Load Region	AREP, Actual LR Size, Max Response Delay, Max Segment Length	—
Initiate Load.rsp(-)	Load Region	AREP, Error Code	—
Pull Segment.rsp(+)	Load Region	AREP, Segment Length, Segment Number, More Follows, Data	—
Pull Segment.rsp(-)	Load Region	AREP, Error Code	—
Push Segment.rsp(+)	Load Region	AREP	—
Push Segment.rsp(-)	Load Region	AREP, Error Code	—
Terminate Load.rsp(+)	Load Region	AREP	—
Terminate Load.rsp(-)	Load Region	AREP, Error Code	—

The parameters used with the primitives are described in the Service Specification of the Load Region object (see 6.2.7.3).

6.2.7.4.1.2.2 Primitives exchanged between the function invocation and the load region state machine

Table 163 and Table 164 show the primitives exchanged between the Function Invocation and the Load Region state machines.

Table 163 – Primitives issued by the Function Invocation to the Load Region state machine

Primitive Name	Source	Associated parameters	Functions
DeleteInUse.ind	Function Invocation	FI Identifier, LR Identifier	The Function Invocation indicates with this primitive that the Load Region Object is no longer to be used.
SetInUse.ind	Function Invocation	FI Identifier, LR Identifier	The Function Invocation indicates with this primitive that it wants to make use of the Load Region Object.

Table 164 – Primitives issued by the Load Region to the Function Invocation state machine

Primitive Name	Source	Associated parameters	Functions
DeleteInUse.rsp (+)	Load Region	FI Identifier, LR Identifier	The Load Region responds with this primitive to a previous received DeleteInUse.ind.
SetInUse.rsp (+/-)	Load Region	FI Identifier, LR Identifier	The Load Region responds with this primitive to a previous received SetInUse.ind.

6.2.7.4.1.3 State definitions

Table 165 shows the definitions of the states of the Load Region state machine.

Table 165 – Load Region state definitions

State	Description
DOWNLOADABLE	This state indicates that the Load Region object is empty, but is capable of being downloaded.
DOWNLOADING	This state indicates that the download sequence has been initiated.
DOWNLOAD SUCCESS	This transient state indicates that the download sequence has succeeded, but has not yet been terminated.
LOADED	This state indicates that the download sequence has terminated successfully.
IN-USE	This state indicates that the Load Region object is loaded and is currently being used by a Function Invocation object.

6.2.7.4.1.4 Local variable definitions

AREP List

The AREP List consists of a one dimensional list containing AREP(s) which represent application relationships who currently execute a download or upload sequence.

Segment Number List

The Segment Number List consists of a two dimensional list containing AREP(s), also contained in the AREP List, and the actual Segment Number of the sequence executed on this relationship.

Actual Length List

The Actual Length List consists of a two dimensional list containing AREP(s), also contained in the AREP List, and the Actual Length of the amount of data already exchanged in the sequence executed on this relationship.

FI Identifier

This local variable contains the Identifier (API, Slot Number, FI Index) of the Function Invocation object which currently uses this Load Region object.

6.2.7.4.1.5 Timer definitions

Intersegment Request Timer

With the Intersegment Request Timer the Load Region state machine supervises the Client activity during an upload or download sequence.

6.2.7.4.1.6 Function definitions

The following functions contained in Table 166 are used by both Load Region state machines which describe the behavior of a Load Region object stored in erasable memory or in non erasable memory.

Table 166 – Load Region function table

Function Name	Description
IsServiceAccepted(Load Type, Load Image Size, Additional Information)	This function checks by calling subfunctions if the service is acceptable and match local object conditions. A) If the results of the called subfunctions are as follows: IsLRSizeOK(Load Image Size) = TRUE IsPushAndUpload() = FALSE IsAddInfoOK(Additional Information) = TRUE than it returns TRUE. B) Otherwise it returns FALSE.
IsPushAndUpload()	This function checks whether an Upload sequence is in progress. A) If the attribute Uploadcount is greater than zero then it returns TRUE. B) Otherwise it returns FALSE.
IsLRSizeOK(Load Image Size)	This function checks whether the Load Image Size is matching the attribute LR Size. A) If the Load Image Size is less or equal LR Size then it returns TRUE. B) Otherwise it returns FALSE.
IsAddInfoOK(Additional Information)	This function checks whether the Additional Information is matching the attribute Load Region Additional Information. A) If the Additional Information is compatible to the attribute Load Region Additional Information then it returns TRUE. B) Otherwise it returns FALSE.
StoreSegmentNumber(AREP, Segment Number)	This function increments the Segment Number with one and stores it AREP related.
GetSegmentNumber(AREP)	This function returns the stored AREP related Segment Number.
CheckSegmentNumber(AREP, Segment Number)	This function compares the Segment Number with the stored AREP related Segment Number. A) If the Segment Number is equal to the stored AREP related Segment Number then it returns TRUE. B) Otherwise it returns FALSE.
StoreActualLength(AREP, Actual Length)	This function stores the Actual Length AREP related.
GetActualLength(AREP)	This function returns the stored AREP related Actual Length.

Function Name	Description
GetActualSegLength(AREP, Segment Length)	This function reads the stored AREP related Actual Length by calling the function GetActualLength and calculates the actual Segment Length according to the following equation: A) If the Actual LR Size-Actual Length is greater or equal to Segment Length and less or equal to Max Segment Length then it returns Segment Length. B) If the Actual LR Size-Actual Length is greater or equal to Segment Length and greater than Max Segment Length then it returns Max Segment Length. C) If the Actual LR Size-Actual Length is less than Segment Length and less or equal to Max Segment Length then it returns Actual LR Size-Actual Length. D) If the Actual LR Size-Actual Length is less than Segment Length and greater Max Segment Length then it returns Max Segment Length. Furthermore this function calculates the new Actual Length according to the following equation: Actual Length = Actual Length + returned value. The new Actual Length is stored AREP related by calling the function StoreActualLength.
CheckActualSegLength(AREP, Segment Length)	This function reads the stored AREP related Actual Length by calling the function GetActualLength and checks the Segment Length according to the following conditions: A) If Segment Length is greater than Max Segment Length then it returns FALSE. B) If Segment Length is greater than Actual LR Size-Actual Length then it returns FALSE. C) Otherwise it returns TRUE.
IsLastSegment(AREP)	This function reads the stored AREP related Actual Length by calling the function GetActualLength and calculates whether it is the last segment to transfer according to the following conditions: A) If the Actual Length is equal to Actual LR Size then it returns FALSE. B) Otherwise it returns TRUE.
StoreAREP(AREP)	This function stores the AREP in the AREP List.
CheckAREP(AREP)	This function checks whether the AREP is contained in the AREP List according to the following conditions: A) If the AREP is contained then it returns TRUE. B) Otherwise it returns FALSE.
DeleteAREP(AREP)	This function deletes the AREP in the AREP List.
StartIntersegReqTimer(AREP, T1)	This function starts or restarts the AREP related Intersegment Request Timer.
StopIntersegReqTimer(AREP)	This function stops the AREP related Intersegment Request Timer.
CheckMoreFollows(AREP; More Follows)	This function reads the stored AREP related Actual Length by calling the function GetActualLength and checks whether it is the last segment to receive according to the following conditions: A) If the Actual Length is equal to Actual LR Size then it returns TRUE. B) Otherwise it returns FALSE.
ClearContent	This function deletes the content of the Load Region object.
StoreFIIdentifier(FI Identifier)	This function stores the FI Identifier.
CheckFIIdentifier(FI Identifier)	This function checks the FI Identifier according to the following conditions: A) If the FI Identifier is equal to the stored FI Identifier then it returns TRUE. B) Otherwise it returns FALSE.
DeleteFIIdentifier(FI Identifier)	This function deletes the stored FI Identifier.
StoreData(Data)	This function stores the Data into the content of the Load Region object.
GetData()	This function gets the Data of the content of the Load Region object.

6.2.7.4.2 Load region state diagrams

6.2.7.4.2.1 Load region state diagram for erasable memory

Figure 25 shows the state diagram of a Load Region for erasable memory.

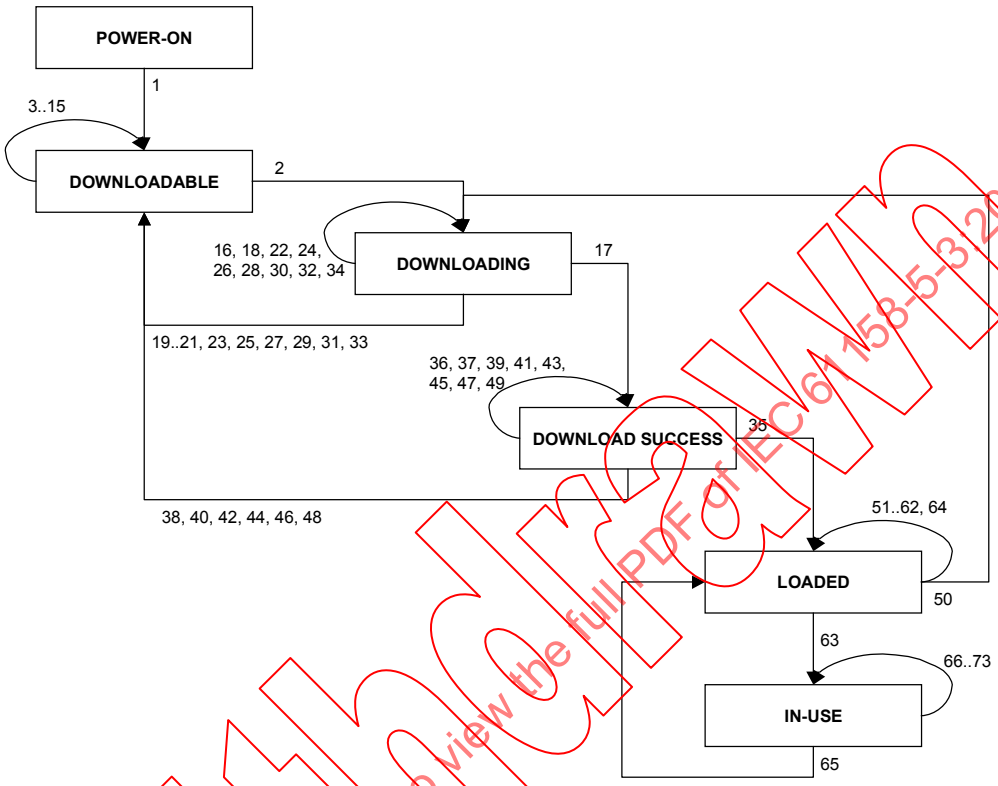

Figure 25 – Load Region state diagram for erasable memory

6.2.7.4.2.2 Load region state diagram for non erasable memory

Figure 26 shows the state diagram of a Load Region for non erasable memory.

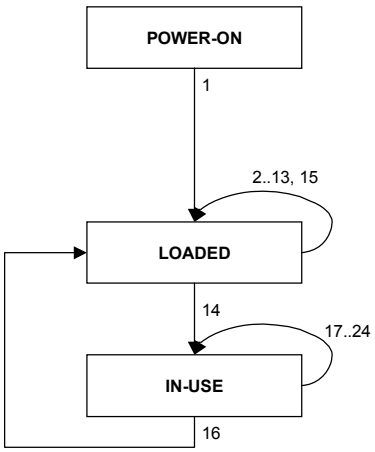

Figure 26 – Load region state diagram for non erasable memory

6.2.7.4.3 Load region state tables

6.2.7.4.3.1 Load region state table for erasable memory

Table 167 shows the state table of a Load Region for erasable memory.

Table 167 – Load Region state table for erasable memory

#	Current State	Event /Condition =>Action	Next State
1	POWER-ON	=> Uploadcount:=0	DOWNLOADABLE
2	DOWNLOADABLE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(Load Type=Push) && (IsServiceAccepted(Load Type, Load Image Size, Additional Information)=TRUE) => StoreAREP(AREP) Segment Number:=0 StoreSegmentNumber(AREP, Segment Number) Actual Length:=0 StoreActualLength(AREP, Actual Length) T1:=Intersegment Request Timeout Actual LR Size:=Load Image Size Load Region State:=DOWNLOADING StartIntersegReqTimer(AREP, T1) Initiate Load.rsp(+) (AREP, Actual LR Size, Max Response Delay, Max Segment Length)	DOWNLOADING

#	Current State	Event /Condition =>Action	Next State
3	DOWNLOADABLE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) && (IsServiceAccepted(Load Type, Additional Information)=TRUE) => T1:=Intersegment Request Timeout Uploadcount:=Uploadcount+1 StoreAREP(AREP) StartIntersegReqTimer(AREP, T1) Initiate Load.rsp(+) (AREP, Actual LR Size, Max Response Delay, Max Segment Length)	DOWNLOADABLE

4	DOWNLOADABLE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information)
		/(Load Type=Push) &&
		(IsServiceAccepted(Load Type, Load Image Size, Additional Information)=FALSE)
		=>
		if (IsPushAndUpload()=TRUE)
		Error Code:=state conflict
		elseif(IsLRSizeOK(Load Image Size)=FALSE)
		Error Code:=load image size error
		Endif
		Initiate Load.rsp(-) (AREP, Error Code)
			DOWNLOADABLE

5	DOWNLOADABLE	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
6	DOWNLOADABLE	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=TRUE => Segment Length:=0 Segment Number:=1 More Follows:=FALSE StartIntersegReqTimer(AREP, T1) Pull Segment.rsp(+) (AREP, Segment Length, Segment Number, More Follows)	DOWNLOADABLE
7	DOWNLOADABLE	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE

#	Current State	Event /Condition =>Action	Next State
8	DOWNLOADABLE	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	DOWNLOADABLE
9	DOWNLOADABLE	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP) Terminate Load.rsp(+) (AREP)	DOWNLOADABLE
10	DOWNLOADABLE	IntersegReqTimer Expired => AREP:=GetTimerAREP() DeleteAREP(AREP) Uploadcount:=Uploadcount-1 IntersegReqTimerExpired.req(AREP)	DOWNLOADABLE
11	DOWNLOADABLE	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP)	DOWNLOADABLE
12	DOWNLOADABLE	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=FALSE =>	DOWNLOADABLE
13	DOWNLOADABLE	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP)	DOWNLOADABLE
14	DOWNLOADABLE	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=FALSE =>	DOWNLOADABLE
15	DOWNLOADABLE	SetInUse.ind (FI Identifier, LR Identifier) => SetInUse.rsp(-) (FI Identifier, LR Identifier)	DOWNLOADABLE

#	Current State	Event /Condition =>Action	Next State
16	DOWNLOADING	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /(CheckAREP(AREP)=TRUE) && (CheckSegmentNumber(AREP, Segment Number)=TRUE) && (CheckActualSegLength(AREP, Segment Length)=TRUE) && (CheckMoreFollows(AREP; More Follows)=TRUE)&& (More Follows=TRUE) => Segment Number:=GetSegmentNumber(AREP) StoreSegmentNumber(AREP, Segment Number) StartIntersegReqTimer(AREP, T1) GetActualSegLength(AREP, Segment Length) StoreData(Data) Push Segment.rsp(+) (AREP)	DOWNLOADING
17	DOWNLOADING	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /(CheckAREP(AREP)=TRUE) && (CheckSegmentNumber(AREP, Segment Number)=TRUE) && (CheckActualSegLength(AREP, Segment Length)=TRUE) && (CheckMoreFollows(AREP; More Follows)=TRUE)&& (More Follows=FALSE) => Segment Number:=GetSegmentNumber(AREP) StoreSegmentNumber(AREP, Segment Number) StartIntersegReqTimer(AREP, T1) GetActualSegLength(AREP, Segment Length) StoreData(Data) Push Segment.rsp(+) (AREP)	DOWNLOAD SUCCESS
18	DOWNLOADING	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /CheckAREP(AREP)=FALSE => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	DOWNLOADING

#	Current State	Event /Condition =>Action	Next State
19	DOWNLOADING	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /(CheckAREP(AREP)=TRUE) && (CheckSegmentNumber(AREP, Segment Number)=FALSE) => Error Code:=invalid parameter DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Push Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
20	DOWNLOADING	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /(CheckAREP(AREP)=TRUE) && (CheckSegmentNumber(AREP, Segment Number)=TRUE) && (CheckActualSegLength(AREP, Segment Length)=FALSE) => Error Code:=length error DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Push Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
21	DOWNLOADING	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /(CheckAREP(AREP)=TRUE) && (CheckSegmentNumber(AREP, Segment Number)=TRUE) && (CheckActualSegLength(AREP, Segment Length)=TRUE) && (CheckMoreFollows(AREP; More Follows)=FALSE) => Error Code:=invalid parameter DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Push Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
22	DOWNLOADING	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	DOWNLOADING

#	Current State	Event /Condition =>Action	Next State
23	DOWNLOADING	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(CheckAREP(AREP)=TRUE) && (Load Type=Push) => Error Code:=state conflict DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Initiate Load.rsp(-) (AREP, Error Code)	DOWNLOADABLE
24	DOWNLOADING	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(CheckAREP(AREP)=FALSE) && (Load Type=Push) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	DOWNLOADING
25	DOWNLOADING	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=TRUE => Error Code:=state conflict DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Pull Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
26	DOWNLOADING	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	DOWNLOADING
27	DOWNLOADING	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=TRUE => Error Code:=state conflict DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Terminate Load.rsp(-) (AREP, Error Code)	DOWNLOADABLE

#	Current State	Event /Condition =>Action	Next State
28	DOWNLOADING	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	DOWNLOADING
29	DOWNLOADING	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0	DOWNLOADABLE
30	DOWNLOADING	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=FALSE =>	DOWNLOADING
31	DOWNLOADING	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0	DOWNLOADABLE
32	DOWNLOADING	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=FALSE =>	DOWNLOADING
33	DOWNLOADING	IntersegReqTimer Expired => AREP:=GetTimerAREP() DeleteAREP(AREP) ClearContent Actual LR Size:=0 IntersegReqTimerExpired.req(AREP)	DOWNLOADABLE
34	DOWNLOADING	SetInUse.ind (FI Identifier, LR Identifier) => SetInUse.rsp(-) (FI Identifier, LR Identifier)	DOWNLOADING
35	DOWNLOAD SUCCESS	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) StopIntersegReqTimer(AREP) Terminate Load.rsp(+) (AREP)	LOADED

#	Current State	Event /Condition =>Action	Next State
36	DOWNLOAD SUCCESS	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	DOWNLOAD SUCCESS
37	DOWNLOAD SUCCESS	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	DOWNLOAD SUCCESS
38	DOWNLOAD SUCCESS	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(CheckAREP(AREP)=TRUE) && (Load Type=Push) => Error Code:=state conflict DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Initiate Load.rsp(-) (AREP, Error Code)	DOWNLOADABLE
39	DOWNLOAD SUCCESS	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(CheckAREP(AREP)=FALSE) && (Load Type=Push) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	DOWNLOAD SUCCESS
40	DOWNLOAD SUCCESS	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=TRUE => Error Code:=state conflict DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Pull Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
41	DOWNLOAD SUCCESS	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	DOWNLOAD SUCCESS

#	Current State	Event /Condition =>Action	Next State
42	DOWNLOAD SUCCESS	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /CheckAREP(AREP)=TRUE => Error Code:=state conflict DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0 Push Segment.rsp(-) (AREP, Error Code)	DOWNLOADABLE
43	DOWNLOAD SUCCESS	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) /CheckAREP(AREP)=FALSE => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	DOWNLOAD SUCCESS
44	DOWNLOAD SUCCESS	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Addittional Detail) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0	DOWNLOADABLE
45	DOWNLOAD SUCCESS	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=FALSE =>	DOWNLOAD SUCCESS
46	DOWNLOAD SUCCESS	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) StopIntersegReqTimer(AREP) ClearContent Actual LR Size:=0	DOWNLOADABLE
47	DOWNLOAD SUCCESS	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=FALSE =>	DOWNLOAD SUCCESS

#	Current State	Event /Condition =>Action	Next State
48	DOWNLOAD SUCCESS	IntersegReqTimer Expired => AREP:=GetTimerAREP() DeleteAREP(AREP) ClearContent Actual LR Size:=0 IntersegReqTimerExpired.req(AREP)	DOWNLOADABLE
49	DOWNLOAD SUCCESS	SetInUse.ind (FI Identifier, LR Identifier) => SetInUse.rsp(-) (FI Identifier, LR Identifier)	DOWNLOAD SUCCESS
50	LOADED	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(Load Type=Push) && (IsServiceAccepted(Load Type, Load Image Size, Additional Information)=TRUE) => StoreAREP(AREP) Segment Number:=0 StoreSegmentNumber(AREP, Segment Number) Actual Length:=0 StoreActualLength(AREP, Actual Length) T1:=Intersegment Request Timeout Actual LR Size:=Load Image Size Load Region State:=DOWNLOADING StartIntersegReqTimer(AREP, T1) Initiate Load.rsp(+) (AREP, Actual LR Size, Max Response Delay, Max Segment Length)	DOWNLOADING
51	LOADED	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) && (IsServiceAccepted(Load Type, Additional Information)=TRUE) => T1:=Intersegment Request Timeout Uploadcount:=Uploadcount+1 StoreAREP(AREP) Segment Number:=0 StoreSegmentNumber(AREP, Segment Number) Actual Length:=0 StoreActualLength(AREP, Actual Length) StartIntersegReqTimer(AREP, T1) Initiate Load.rsp(+) (AREP, Actual LR Size, Max Response Delay, Max Segment Length)	LOADED

#	Current State	Event /Condition =>Action	Next State

52	LOADED	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information)
		/(Load Type=Push) &&
		(IsServiceAccepted(Load Type, Load Image Size, Additional Information)=FALSE)
		=>
		if (IsPushAndUpload()=TRUE)
		Error Code:=state conflict
		Elseif(IsLRSizeOK(Load Image Size)=FALSE)
		Error Code:=load image size error
		Endif
		Initiate Load.rsp(-) (AREP, Error Code)
			LOADED

53	LOADED	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	LOADED
54	LOADED	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=TRUE => Segment Number:=GetSegmentNumber(AREP) StoreSegmentNumber(AREP, Segment Number) Segment Length:=GetActualSegLength(AREP, Segment Length) More Follows:=IsLastSegment(AREP) StartIntersegReqTimer(AREP, T1) Data:=GetData() Pull Segment.rsp(+) (AREP, Segment Length, Segment Number, More Follows, Data)	LOADED
55	LOADED	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	LOADED
56	LOADED	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP) Terminate Load.rsp(+) (AREP)	LOADED
57	LOADED	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	LOADED

#	Current State	Event /Condition =>Action	Next State
58	LOADED	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP)	LOADED
59	LOADED	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=FALSE =>	LOADED
60	LOADED	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP)	LOADED
61	LOADED	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=FALSE =>	LOADED
62	LOADED	IntersegReqTimer Expired => AREP:=GetTimerAREP() DeleteAREP(AREP) Uploadcount:=Uploadcount-1 IntersegReqTimerExpired.req(AREP)	LOADED
63	LOADED	SetInUse.ind (FI Identifier, LR Identifier) /Uploadcount=0 => StoreFIIdentifier(FI Identifier) SetInUse.rsp(+) (FI Identifier, LR Identifier)	IN-USE
64	LOADED	SetInUse.ind (FI Identifier, LR Identifier) /Uploadcount>0 => SetInUse.rsp(-) (FI Identifier, LR Identifier)	LOADED
65	IN-USE	DeleteInUse.ind (FI Identifier, LR Identifier) => DeleteFIIdentifier(FI Identifier) DeleteInUse.rsp(+)(FI Identifier, LR Identifier)	LOADED
66	IN-USE	SetInUse.ind (FI Identifier, LR Identifier) /CheckFIIdentifier(FI Identifier)=FALSE => SetInUse.rsp(-) (FI Identifier, LR Identifier)	IN-USE

#	Current State	Event /Condition =>Action	Next State
67	IN-USE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(Load Type=Push) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	IN-USE
68	IN-USE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	IN-USE
69	IN-USE	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	IN-USE
70	IN-USE	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	IN-USE
71	IN-USE	Terminate Load.ind (AREP, Slot Number, LR Index) => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	IN-USE
72	IN-USE	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) =>	IN-USE
73	IN-USE	DP-slave Stopped.ind (AREP) =>	IN-USE

6.2.7.4.4 Load region state table for non erasable memory

Table 168 shows the state table of a Load Region for non erasable memory.

Table 168 – Load Region state table for non erasable memory

#	Current State	Event /Condition =>Action	Next State
1	POWER-ON	=> Uploadcount:=0	LOADED
2	LOADED	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(Load Type=Push) => Error Code:= access denied Initiate Load.rsp(-) (AREP, Error Code)	LOADED
3	LOADED	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) && (IsServiceAccepted(Load Type, Additional Information)=TRUE) => T1:=Intersegment Request Timeout Uploadcount:=Uploadcount+1 StoreAREP(AREP) Segment Number:=0 StoreSegmentNumber(AREP, Segment Number) Actual Length:=0 StoreActualLength(AREP, Actual Length) StartIntersegReqTimer(AREP, T1) Initiate Load.rsp(+) (AREP, Actual LR Size, Max Response Delay, Max Segment Length)	LOADED
4	LOADED	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	LOADED
5	LOADED	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=TRUE => Segment Number:=GetSegmentNumber(AREP) StoreSegmentNumber(AREP, Segment Number) Segment Length:=GetActualSegLength(AREP, Segment Length) More Follows:=IsLastSegment(AREP) StartIntersegReqTimer(AREP, T1) Data:=GetData() Pull Segment.rsp(+) (AREP, Segment Length, Segment Number, More Follows, Data)	LOADED

#	Current State	Event /Condition =>Action	Next State
6	LOADED	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	LOADED
7	LOADED	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP) Terminate Load.rsp(+) (AREP)	LOADED
8	LOADED	Terminate Load.ind (AREP, Slot Number, LR Index) /(CheckAREP(AREP)=FALSE => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	LOADED
9	LOADED	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP)	LOADED
10	LOADED	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) /(CheckAREP(AREP)=FALSE =>	LOADED
11	LOADED	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=TRUE => DeleteAREP(AREP) Uploadcount:=Uploadcount-1 StopIntersegReqTimer(AREP)	LOADED
12	LOADED	DP-slave Stopped.ind (AREP) /(CheckAREP(AREP)=FALSE =>	LOADED
13	LOADED	IntersegReqTimer Expired => AREP:=GetTimerAREP() DeleteAREP(AREP) Uploadcount:=Uploadcount-1 IntersegReqTimerExpired.req(AREP)	LOADED

#	Current State	Event /Condition =>Action	Next State
14	LOADED	SetInUse.ind (FI Identifier, LR Identifier) /Uploadcount=0 => StoreFIIdentifier(FI Identifier) SetInUse.rsp(+) (FI Identifier, LR Identifier)	IN-USE
15	LOADED	SetInUse.ind (FI Identifier, LR Identifier) /Uploadcount>0 => SetInUse.rsp(-) (FI Identifier, LR Identifier)	LOADED
16	IN-USE	DeleteInUse.ind (FI Identifier, LR Identifier) => DeleteFIIdentifier(FI Identifier) DeleteInUse.rsp(+)(FI Identifier, LR Identifier)	LOADED
17	IN-USE	SetInUse.ind (FI Identifier, LR Identifier) /CheckFIIdentifier(FI Identifier)=FALSE => SetInUse.rsp(-) (FI Identifier, LR Identifier)	IN-USE
18	IN-USE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Load Image Size, Intersegment Request Timeout, Additional Information) /(Load Type=Push) => Error Code:= access denied Initiate Load.rsp(-) (AREP, Error Code)	IN-USE
19	IN-USE	Initiate Load.ind (AREP, Slot Number, LR Index, Load Type, Intersegment Request Timeout, Additional Information) /(Load Type=Pull) => Error Code:=state conflict Initiate Load.rsp(-) (AREP, Error Code)	IN-USE
20	IN-USE	Push Segment.ind (AREP, Slot Number, LR Index, Segment Length, Segment Number, More Follows, Data) => Error Code:=state conflict Push Segment.rsp(-) (AREP, Error Code)	IN-USE
21	IN-USE	Pull Segment.ind (AREP, Slot Number, LR Index, Segment Length) => Error Code:=state conflict Pull Segment.rsp(-) (AREP, Error Code)	IN-USE
22	IN-USE	Terminate Load.ind (AREP, Slot Number, LR Index) => Error Code:=state conflict Terminate Load.rsp(-) (AREP, Error Code)	IN-USE

#	Current State	Event /Condition =>Action	Next State
23	IN-USE	Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail) =>	IN-USE
24	IN-USE	DP-slave Stopped.ind (AREP) =>	IN-USE

6.2.8 Function invocation ASE

6.2.8.1 Overview

The Function Invocation ASE defines two objects, the stateless Action object and the state-oriented Function Invocation object.

Stateless function invocations, referred to as actions, run to completion when invoked and cannot be interrupted. In addition, some atomic actions return a value in response to being invoked, while others do not. Those that do may be used to model software *functions*, and those that do not may be used to model software *procedures*.

State-oriented function invocations, on the other hand, may be controlled during their execution. Services are defined to stop, resume, or reset them once they have been invoked. They do not return a value in response to being started. If it is necessary to stop a function invocation once it has been started, then it shall be defined as state-oriented.

State-oriented function invocations represent the network view of a specific invocation of a User function. If the User function can be invoked more than once simultaneously, then separate Function Invocation objects are needed to represent each invocation.

State-oriented function invocations may be used to model software processes or user operations that may be started and controlled.

NOTE An example of a state-oriented function invocation that may be used to model a user operation is the “playback” operation of a video recorder. Once it has been started, the playback operation may be stopped (paused) and later resumed.

To support the concept of software process modeling, the definition of functional invocations includes optional attributes that can be used to relate them to Load Region objects.

The Function Invocation objects shall clearly be separated from the Process Data objects and the Load Region objects. No overlapping is allowed.

6.2.8.2 Function invocation model class specifications

6.2.8.2.1 Function invocation class definition

6.2.8.2.1.1 Function invocation formal model

A Function Invocation object is described by the following template:

DP ASE: **Function Invocation ASE**
CLASS: Function Invocation
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Identifier
3.	(m)	Attribute:	Access Rights MS1
4.	(m)	Attribute:	Access Rights MS2
4.1.	(m)	Attribute:	List of SCL

4.1.1.	(m)	Attribute:	SCL
7.	(m)	Attribute:	Function Invocation State
8.	(o)	Attribute:	Load Region Object Identifier
8.1	(c)	Attribute:	Load Region Object In Use
9.	(o)	Attribute:	Execution Argument Data Description
10.	(o)	Attribute:	Function Invocation Name
11.	(o)	Attribute:	Local Detail

SERVICES:

1.	(m)	OpsService:	Start
2.	(o)	OpsService:	Stop
3.	(o)	OpsService:	Resume
4.	(m)	OpsService:	Reset
5.	(m)	OpsService:	Get FI State

6.2.8.2.1.2 Attributes

Identifier
This key attribute is a triplet composed of an API, Slot Number, and FI Index to define to which Application Process Identifier, module and to which index within the specified module the Function Invocation object belongs. This Identifier shall not be used by another Action object, Function Invocation object, Load Region object or Process Data object.

Attribute Type: Unsigned8, Unsigned8, Unsigned16

Allowed triplet values: [0 to 254, 0 to 254, 0 to 65 199 (65 200 to 65 535 reserved)]

Access Rights MS1
This attribute defines the type of access rights defined for a MS1 AR for the Function Invocation object.

Attribute Type: Unsigned8

The allowed values are shown in Table 169.

Table 169 – Access rights MS1

Value	Meaning
2	Right to Start, Stop, Resume, Reset
3	No Access

Access Rights MS2
This attribute defines the type of access rights defined for a MS2 AR for the Function Invocation object.

Attribute Type: Unsigned8

The allowed values are shown in Table 170.

Table 170 – Access rights MS2

Value	Meaning
2	Right to Start, Stop, Resume, Reset
3	No Access

List of SCL
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one value for access protection to this Function Invocation object. The rules for accepting or rejecting an access are described in 6.2.1.3.

Attribute Type: Unsigned8

Function Invocation State
This attribute specifies the state of the Function Invocation object.

Attribute Type: Unsigned8

The allowed values are shown in Table 171.

Table 171 – Function Invocation state

Value	Meaning
0	IDLE
1	STARTING
2	RUNNING
3	STOPPING
4	STOPPED
5	RESUMING
6	UNRUNNABLE
7	RESETTING
8	WAIT-SET-IN-USE
9	WAIT-SET-IN-USE-2
10	WAIT-DEL-IN-USE
11	WAIT-DEL-IN-USE-2
12	EXEC-ERR
13	EXEC-TERM

Load Region Object Identifier
This optional attribute is a triplet composed of API, Slot Number and LR Index to define to which Application Process Identifier, module and to which index within the specified module the related Load Region object belongs.

Attribute Type: Unsigned8, Unsigned8, Unsigned16

Allowed triplet values: [0 to 254, 0 to 254, 0 to 65 199 (65 200 to 65 535 reserved)]

Load Region Object In Use
This Boolean attribute indicates whether the related Load Region object is actual used by this Function Invocation object.

Attribute Type: Boolean

The allowed values are shown in Table 172.

Table 172 – Load Region object in use

Value	Meaning
FALSE	Load Region object not in use
TRUE	Load Region object in use

Execution Argument Data Description
This optional attribute contains the Data Description for the Parameter Execution Argument of the Start service. This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Function Invocation Name
This optional attribute defines the name of the Function Invocation object.

Attribute Type: Visible String(32)

Local Detail
This optional attribute contains additional local information about the Function Invocation object.

Attribute Type: Octet String

6.2.8.2.1.3 Invocation of the function invocation object

For the invocation of the Function Invocation object the following rules apply:

— At least one AR shall have access to the Function Invocation object.
— Only one Load Region object shall be related to a Function Invocation object.
— The access rights shall be set accordingly to the allowed service(s).
— The optional services shall be selected according to the execution mode (cyclic, non cyclic, …) of the function invocation.

6.2.8.2.2 Action class definition

6.2.8.2.2.1 Action formal model

An Action object is described by the following template:

DP ASE:			**Function Invocation ASE**
CLASS:		Action	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Identifier
3.	(m)	Attribute:	Access Rights MS1
4.	(m)	Attribute:	Access Rights MS2
4.1.	(m)	Attribute:	List of SCL
4.1.1.	(m)	Attribute:	SCL
8.	(o)	Attribute:	Load Region Object Identifier
8.1	(c)	Attribute:	Load Region Object In Use
9.	(o)	Attribute:	Execution Argument Data Description
10.	(o)	Attribute:	Result Argument Data Description
11.	(o)	Attribute:	Action Name
12.	(o)	Attribute:	Local Detail
SERVICES:			
1.	(m)	OpsService:	Call

6.2.8.2.2.2 Attributes

Identifier
This key attribute is a triplet composed of API, Slot Number and FI Index to define to which Application Process Identifier, module and to which index within the specified module the Action object belongs. This Identifier shall not be used by another Action object, Function Invocation object, Load Region object or Process Data object.

Attribute Type: Unsigned8, Unsigned8, Unsigned16

Allowed triplet values: [0 to 254, 0 to 254, 0 to 65 199 (65 200 to 65 535 reserved)]

Access Rights MS1
This attribute defines the type of access rights defined for a MS1 AR for the Action object.

Attribute Type: Unsigned8

The allowed values are shown in Table 173.

Table 173 – Access rights MS1

Value	Meaning
2	Right to Call
3	No Access

Access Rights MS2
This attribute defines the type of access rights defined for a MS2 AR for the Action object.

Attribute Type: Unsigned8

The allowed values are shown in Table 174.

Table 174 – Access rights MS2

Value	Meaning
2	Right to Call
3	No Access

List of SCL
This attribute is composed of the following list elements:

SCL
The attribute SCL contains one value for access protection to this Action object. The rules for accepting or rejecting an access are described in 6.2.1.3.

Attribute Type: Unsigned8

Load Region Object Identifier
This optional attribute is a triplet composed of API, Slot Number and LR Index to define to which Application Process Identifier, module and to which index within the specified module the related Load Region object belongs.

Attribute Type: Unsigned8, Unsigned8, Unsigned16

Allowed triplet values: [0 to 254, 0 to 254, 0 to 65 199 (65 200 to 65 535 reserved)]

Load Region Object In Use
This Boolean attribute indicates whether the related Load Region object is actual used by this Action object.

Attribute Type: Boolean

The allowed values are shown in Table 175.

Table 175 – Load Region object in use

Value	Meaning
FALSE	Load Region object not in use
TRUE	Load Region object in use

Execution Argument Data Description
This optional attribute contains the Data Description for the Parameter Execution Argument of the request of the Call service. This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Result Argument Data Description
This optional attribute contains the Data Description for the Parameter Execution Argument of the response of the Call service. This attribute may contain either a Simple Data Description or an Array Data Description or a Record Data Description. For data description the template given in IEC 61158-5-10, 5.1 is used.

Action Name
This optional attribute defines the name of the Action object.

Attribute Type: Visible String(32)

Local Detail
This optional attribute contains additional local information about the Action object.

Attribute Type: Octet String

6.2.8.2.2.3 Invocation of the action object

For the invocation of the Action object the following rules apply:

— At least one AR shall have access to the Action object.
— Only one Load Region object shall be related to an Action object.
— The access rights shall be set accordingly to the allowed service(s).

6.2.8.3 Function invocation service specification

6.2.8.3.1 Start service

This confirmed service is used to request that a Function Invocation be started. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 176 shows the parameters of the service.

Table 176 – Start

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
FI Index	M	M(=)		
Execution Argument	U	U(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Function Invocation object in a specific slot (typically a module).

FI Index
This parameter specifies the index of the Function Invocation to be started.

Execution Argument
This optional parameter contains additional information to perform the start of the Function Invocation. It shall be present if the attribute Execution Argument Data Description is defined for the addressed Function Invocation object.

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

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: state conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type, resource busy.

6.2.8.3.2 Stop service

This confirmed service is used to stop a Function Invocation retaining its context so that it may be resumed. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 177 shows the parameters of the service.

Table 177 – Stop

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
FI Index	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Function Invocation object in a specific slot (typically a module).

FI Index
This parameter specifies the index of the Function Invocation to be stopped.

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

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: state conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type, feature not supported.

6.2.8.3.3 Resume service

This confirmed service is used to resume the execution of a Function Invocation that has been stopped. The execution is resumed using the context saved when the Function Invocation was stopped. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 178 shows the parameters of the service.

Table 178 – Resume

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
FI Index	M	M(=)		
Execution Argument	U	U(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Function Invocation object in a specific slot (typically a module).

FI Index
This parameter specifies the index of the Function Invocation to be resumed.

Execution Argument
This optional parameter contains additional information to resume the execution of the Function Invocation. It shall be present if the attribute Execution Argument Data Description is defined for the addressed Function Invocation object.

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

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: state conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type, feature not supported, resource busy.

6.2.8.3.4 Reset service

This confirmed service is used to reset a Function Invocation with its initial context. Its initial context is defined as the context of the Function Invocation set by its initialization procedures. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 179 shows the parameters of the service.

Table 179 – Reset

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
FI Index	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Function Invocation object in a specific slot (typically a module).

FI Index
This parameter specifies the index of the Function Invocation to be reset.

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

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: state conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type.

6.2.8.3.5 Get FI state service

This confirmed service is used to read the actual state of a Function Invocation object. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 180 shows the parameters of the service.

Table 180 – Get FI state

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
FI Index	M	M(=)		
Result(+)			S	S(=)
AREP			M	M(=)
FI State			M	M(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Function Invocation object in a specific slot (typically a module).

FI Index
This parameter specifies the index of the Function Invocation to be stopped.

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

FI State
This parameter contains the actual state of the Function Invocation object.

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type.

6.2.8.3.6 Call service

This confirmed service is used to invoke the execution of an Action object. This service shall only be used in conjunction with the MS1 or MS2 AR.

Table 181 shows the parameters of the service.

Table 181 – Call

Parameter name	Req	Ind	Rsp	Cnf
Argument	M	M(=)		
AREP	M	M(=)		
Slot Number	M	M(=)		
Entity Number	U	U(=)		
FI Index	M	M(=)		
Execution Argument	U	U(=)		
Result(+)			S	S(=)
AREP			M	M(=)
Result Argument			U	U(=)
Result(-)			S	S(=)
AREP			M	M(=)
Error Code			M	M(=)

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Slot Number
The parameter Slot Number is used in the destination device for addressing the desired Action object in a specific slot (typically a module).

Entity Number
The parameter Entity Number is profile specific.

FI Index
This parameter specifies the index of the Action object to be invoked. The available index range is structured as follows:

- Indices reserved for manufacturer specific use (0 - 32 767)
- Indices reserved for PROFIBUS system services (32 768 - 64 999)
- Indices reserved for Identification and Maintenance (I&M) services (65 000 - 65 199)

Execution Argument
This optional parameter contains additional information to perform the invocation of the Action object. It shall be present if the attribute Execution Argument Data Description is defined for the addressed Action object.

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

Result Argument
This optional parameter contains the result of the invocation of the Action object. It shall be present if the attribute Result Argument Data Description is defined for the addressed Action object.

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

Error Code
This parameter indicates the reason for the failure.

Allowed values: type conflict, access denied, invalid index, invalid slot, module failure, invalid parameter, invalid type.

6.2.8.4 Behavior of the function invocation object

6.2.8.4.1 Function invocation state machine description

6.2.8.4.1.1 General

The behavior of a Function Invocation object is defined by the Function Invocation state machine. Each Function Invocation object is represented by an own instance of this state machine.

The Function Invocation state machine represents a part of the User functionality and is included to ensure interoperability. It is assumed that other parts of the User functionality perform the following checks:

— validation of existence and addressing of the Function Invocation object
— validation of the access rights
— validation of temporary system restrictions for the use of the Function Invocation object
— validation of the correct class of the object

and generate appropriate Error Codes if necessary.

The Function Invocation object may make use of a Load Region object. Therefore the Function Invocation state machine provides an interface to the Load Region state machines for co-ordination purposes.

The main functionalities of the Function Invocation state machine are:

— checking of the validity of the services versus the state of the Function Invocation object
— to ensure that no parallel services are executed on the Function Invocation object
— to ensure that a Function Invocation object is only used if the related Load Region object is available
— supervision of the Client activity.

6.2.8.4.1.2 Primitive definitions

6.2.8.4.1.2.1 Primitives exchanged between the user and the function invocation state machine

Table 182 and Table 183 show the primitive exchanged between the User and the Function Invocation state machine.

Table 182 – Primitives issued by the user to the Function Invocation state machine

Primitive Name	Source	Associated parameters	Functions
Abort.ind	User	AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail	—
Call.ind	User	AREP, Slot Number, FI Index, Execution Argument	—
DP-slave Stopped.ind	User	AREP	—
Reset.ind	User	AREP, Slot Number, FI Index	—
Resume.ind	User	AREP, Slot Number, FI Index, Execution Argument	—
Start.ind	User	AREP, Slot Number, FI Index, Execution Argument	—
Stop.ind	User	AREP, Slot Number, FI Index	—
Get FI State.ind	User	AREP, Slot Number, FI Index	—

Table 183 – Primitives issued by the Function Invocation state machine to the user

Primitive Name	Source	Associated parameters	Functions
Reset.rsp(+)	Function Invocation	AREP	—
Reset.rsp(-)	Function Invocation	AREP, Error Code, Function Invocation State	—
Resume.rsp(+)	Function Invocation	AREP	—
Resume.rsp(-)	Function Invocation	AREP, Error Code, Function Invocation State	—
Start.rsp(+)	Function Invocation	AREP	—
Start.rsp(-)	Function Invocation	AREP, Error Code, Function Invocation State	—
Stop.rsp(+)	Function Invocation	AREP	—
Stop.rsp(-)	Function Invocation	AREP, Error Code, Function Invocation State	—
Get FI State.rsp(+)	Function Invocation	AREP, FI State	—

The parameters used with the primitives are described in the Service Specification of the Function Invocation object (see 6.2.8.3).

6.2.8.4.1.2.2 Primitives exchanged between the LR and the function invocation state machine

Table 184 and Table 185 show the primitive exchanged between the Load Region state machine and the Function Invocation state machine.

Table 184 – Primitives issued by the Load Region to the Function Invocation state machine

Primitive Name	Source	Associated parameters	Functions
DeleteInUse.cnf(+)	Load Region	FI Identifier, LR Identifier	The Load Region confirms with this primitive positively a previous issued DeleteInUse.req.
SetInUse.cnf(+)	Load Region	FI Identifier, LR Identifier	The Load Region confirms with this primitive positively a previous issued SetInUse.req.
SetInUse.cnf(-)	Load Region	FI Identifier, LR Identifier	The Load Region confirms with this primitive negatively a previous issued SetInUse.req.

Table 185 – Primitives issued by the Function Invocation to the Load Region state machine

Primitive Name	Source	Associated parameters	Functions
DeleteInUse.req	Function Invocation	FI Identifier, LR Identifier	The Function Invocation requests with this primitive the termination of the use of a Load Region object.
SetInUse.req	Function Invocation	FI Identifier, LR Identifier	The Function Invocation requests with this primitive the use of a Load Region object.

6.2.8.4.1.3 State definitions

Table 186 shows the definitions of the states of the Load Region state machine.

Table 186 – Function Invocation state definitions

State	Description
IDLE	This state indicates that the Function Invocation object is not used and can be started.
STARTING	This transient state indicates that the Function Invocation is executing a start procedure.
RUNNING	This state indicates that the object has completed the start-up procedure and executes its normal function.
STOPPING	This transient state indicates that a previous issued Stop is executed.
STOPPED	This state indicates that the Function Invocation is stopped.
RESUMING	This transient state indicates that a Resume command is executed.
UNRUNNABLE	This state indicates that an error occurred.
RESETTING	This transient state indicates that a Reset Command is executed.
WAIT-SET-IN-USE	This transient state indicates that the Function Invocation is executing a start procedure after breakdown of the related Association Relationship.
WAIT-SET-IN-USE-2	This transient state indicates that the Function Invocation is executing a resume procedure after breakdown of the related Association Relationship.
WAIT-DEL-IN-USE	This transient state indicates that the Function Invocation is executing a stop procedure after breakdown of the related Association Relationship.
WAIT-DEL-IN-USE-2	This transient state indicates that the Function Invocation is executing a reset procedure after breakdown of the related Association Relationship.
EXEC-ERR	This transient state indicates that during the execution of a Function Invocation an error has been detected.
EXEC-TERM	This transient state indicates that the execution of a Function Invocation is terminated at the end of the program in case of non cyclic execution mode.

6.2.8.4.1.4 Local variable definitions

AREP

This local variable contains the AREP which represents the application relationships who currently executes a Start, Stop, Resume or Reset service.

LR Identifier

This local variable contains the Identifier (API, Slot Number, LR Index) of the Load Region object which currently is used by this Function Invocation object.

6.2.8.4.1.5 Local event definitions

Execution Error
This local event is generated if during the execution of a Function Invocation an error has been detected. The execution is terminated and the Function Invocation is set to the state UNRUNNABLE.

Execution Terminated
This local event is generated if the execution of a Function Invocation is terminated at the end of the program in case of non cyclic execution modes. The Function Invocation is set to the state IDLE.

6.2.8.4.1.6 Function definitions

The functions contained in Table 187 are used by both Function Invocation state machines which describe the behavior of a Function Invocation object.

Table 187 – Function definitions

Function Name	Description
IsServiceAccepted(Execution Argument)	This function checks if the received Execution Argument is compatible to the corresponding attribute of the Function Invocation object. A) If the received Execution Argument is compatible then it returns TRUE. B) Otherwise it returns FALSE.
CheckLRIdentifier (LR Identifier)	This function checks if the received LR Identifier is equal to the corresponding attribute of the Function Invocation object. A) If the received LR Identifier is equal then it returns TRUE. B) Otherwise it returns FALSE.
StoreAREP(AREP)	This function stores the received AREP in the variable AREP.
CheckAREP(AREP)	This function checks if the received AREP is equal to the AREP stored in the variable AREP. A) If the received AREP is equal then it returns TRUE. B) Otherwise it returns FALSE.
GetAREP ()	This function reads the AREP stored in the variable AREP.
GetFIState()	This function reads the actual state from the attribute Function Invocation State of the Function Invocation object.

6.2.8.4.2 Function invocation state diagram

Figure 27 shows the state diagram of the Function Invocation state machine.

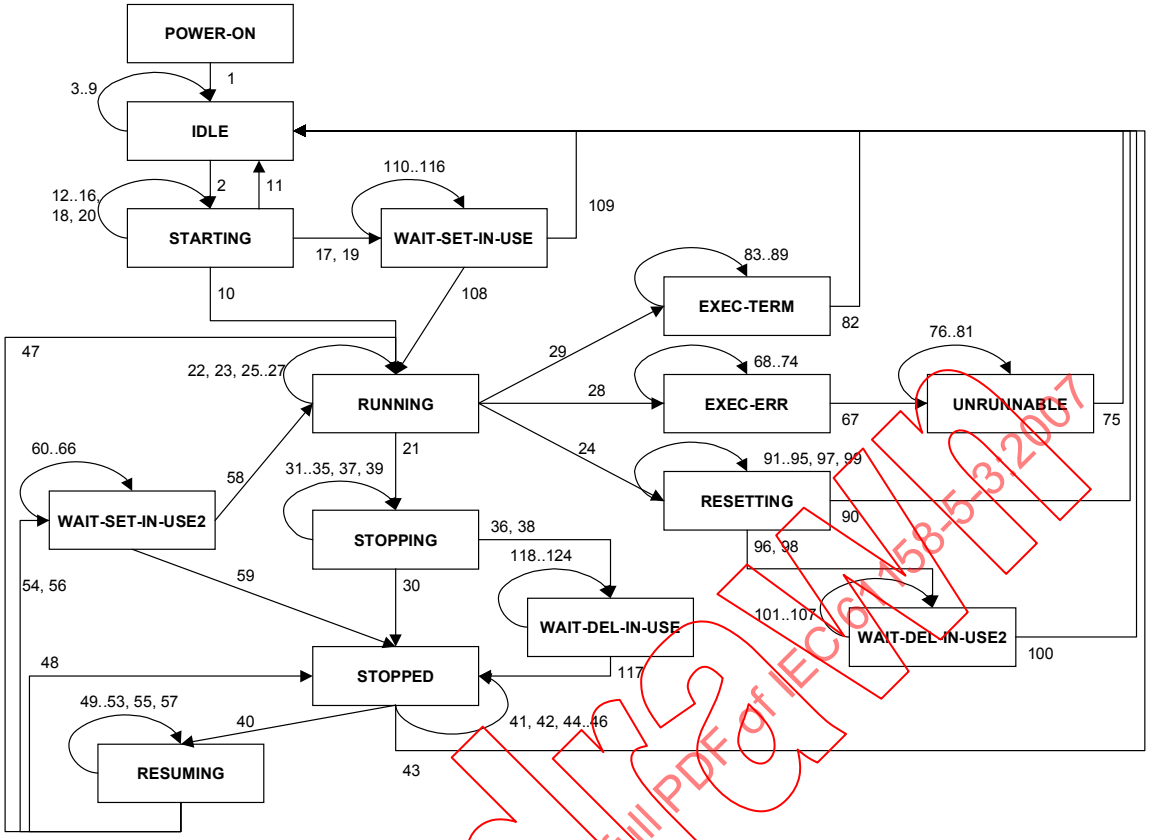

Figure 27 – Function invocation state diagram

6.2.8.4.3 Function invocation state table

Table 188 shows the states of a Function Invocation state machine.

Table 188 – Function Invocation state table

#	Current State	Event /Condition =>Action	Next State
1	POWER-ON	=>	IDLE
2	IDLE	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** /(IsServiceAccepted(Execution Argument) = TRUE) => StoreAREP(AREP) SetInUse.req (FI Identifier, LR Identifier)	STARTING
3	IDLE	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** /(IsServiceAccepted(Execution Argument) = FALSE) => Error Code := invalid parameter Start.rsp(-) (AREP, Error Code)	IDLE

#	Current State	Event /Condition =>Action	Next State
4	IDLE	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	IDLE
5	IDLE	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	IDLE
6	IDLE	**Reset.ind (AREP, Slot Number, FI Index)** => Reset.rsp (+) (AREP)	IDLE
7	IDLE	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	IDLE
8	IDLE	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	IDLE
9	IDLE	**DP-slave Stopped.ind (AREP)** =>	IDLE
10	STARTING	**SetInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) => AREP := GetAREP () Start.rsp(+) (AREP)	RUNNING
11	STARTING	**SetInUse.cnf (-) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) => AREP := GetAREP () Error Code := resource busy Start.rsp(-) (AREP, Error Code)	IDLE
12	STARTING	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	STARTING

#	Current State	Event /Condition =>Action	Next State
13	STARTING	**Stop.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	STARTING
14	STARTING	**Resume.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	STARTING
15	STARTING	**Reset.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	STARTING
16	STARTING	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	STARTING
17	STARTING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = TRUE) =>	WAIT-SET-IN-USE
18	STARTING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = FALSE) =>	STARTING
19	STARTING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = TRUE) =>	WAIT-SET-IN-USE
20	STARTING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = FALSE) =>	STARTING
21	RUNNING	**Stop.ind (AREP, Slot Number, FI Index)** => StoreAREP(AREP) DeleteInUse.req (FI Identifier, LR Identifier)	STOPPING

#	Current State	Event /Condition =>Action	Next State
22	RUNNING	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	RUNNING
23	RUNNING	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	RUNNING
24	RUNNING	**Reset.ind (AREP, Slot Number, FI Index)** => DeleteInUse.req (FI Identifier, LR Identifier)	RESETTING
25	RUNNING	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	RUNNING
26	RUNNING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	RUNNING
27	RUNNING	**DP-slave Stopped.ind (AREP)** =>	RUNNING
28	RUNNING	**Execution Error** => DeleteInUse.req (FI Identifier, LR Identifier)	EXEC-ERR
29	RUNNING	**Execution Terminated** => DeleteInUse.req (FI Identifier, LR Identifier)	EXEC-TERM
30	STOPPING	**DeleteInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) => AREP := GetAREP () Stop.rsp(+) (AREP)	STOPPED
31	STOPPING	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	STOPPING

#	Current State	Event /Condition =>Action	Next State
32	STOPPING	**Stop.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	STOPPING
33	STOPPING	**Resume.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	STOPPING
34	STOPPING	**Reset.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	STOPPING
35	STOPPING	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	STOPPING
36	STOPPING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = TRUE) =>	WAIT-DEL-IN-USE
37	STOPPING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = FALSE) =>	STOPPING
38	STOPPING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = TRUE) =>	WAIT-DEL-IN-USE
39	STOPPING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = FALSE) =>	STOPPING
40	STOPPED	**Resume.ind (AREP, Slot Number, FI Index)** => StoreAREP(AREP) SetInUse.req (FI Identifier, LR Identifier)	RESUMING

#	Current State	Event /Condition =>Action	Next State
41	STOPPED	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	STOPPED
42	STOPPED	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	STOPPED
43	STOPPED	**Reset.ind (AREP, Slot Number, FI Index)** => Reset.rsp (+) (AREP)	IDLE
44	STOPPED	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	STOPPED
45	STOPPED	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	STOPPED
46	STOPPED	**DP-slave Stopped.ind (AREP)** =>	STOPPED
47	RESUMING	**SetInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) => AREP := GetAREP () Resume.rsp(+) (AREP)	RUNNING
48	RESUMING	**SetInUse.cnf (-) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) => AREP := GetAREP () Error Code := resource busy Resume.rsp(-) (AREP, Error Code)	STOPPED
49	RESUMING	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	RESUMING

#	Current State	Event /Condition =>Action	Next State
50	RESUMING	**Stop.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	RESUMING
51	RESUMING	**Resume.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	RESUMING
52	RESUMING	**Reset.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	RESUMING
53	RESUMING	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	RESUMING
54	RESUMING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = TRUE) =>	WAIT-SET-IN-USE-2
55	RESUMING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = FALSE) =>	RESUMING
56	RESUMING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = TRUE) =>	WAIT-SET-IN-USE-2
57	RESUMING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = FALSE) =>	RESUMING
58	WAIT-SET-IN-USE-2	**SetInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	RUNNING

#	Current State	Event /Condition =>Action	Next State
59	WAIT-SET-IN-USE-2	**SetInUse.cnf (-) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	STOPPED
60	WAIT-SET-IN-USE-2	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	WAIT-SET-IN-USE-2
61	WAIT-SET-IN-USE-2	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	WAIT-SET-IN-USE-2
62	WAIT-SET-IN-USE-2	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	WAIT-SET-IN-USE-2
63	WAIT-SET-IN-USE-2	**Reset.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	WAIT-SET-IN-USE-2
64	WAIT-SET-IN-USE-2	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	WAIT-SET-IN-USE-2
65	WAIT-SET-IN-USE-2	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	WAIT-SET-IN-USE-2
66	WAIT-SET-IN-USE-2	**DP-slave Stopped.ind (AREP)** =>	WAIT-SET-IN-USE-2
67	EXEC-ERR	**DeleteInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	UNRUNNABLE
68	EXEC-ERR	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	EXEC-ERR

#	Current State	Event /Condition =>Action	Next State
69	EXEC-ERR	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	EXEC-ERR
71	EXEC-ERR	**Reset.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	EXEC-ERR
72	EXEC-ERR	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	EXEC-ERR
73	EXEC-ERR	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	EXEC-ERR
74	EXEC-ERR	**DP-slave Stopped.ind (AREP)** =>	EXEC-ERR
75	UNRUNNABLE	**Reset.ind (AREP, Slot Number, FI Index)** => Reset.rsp (+) (AREP)	IDLE
76	UNRUNNABLE	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	UNRUNNABLE
77	UNRUNNABLE	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	UNRUNNABLE
78	UNRUNNABLE	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	UNRUNNABLE
79	UNRUNNABLE	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	UNRUNNABLE
80	UNRUNNABLE	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	UNRUNNABLE

#	Current State	Event /Condition =>Action	Next State
81	UNRUNNABLE	**DP-slave Stopped.ind (AREP)** =>	UNRUNNABLE
82	EXEC-TERM	**DeleteInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	IDLE
83	EXEC-TERM	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	EXEC-TERM
84	EXEC-TERM	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	EXEC-TERM
85	EXEC-TERM	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	EXEC-TERM
86	EXEC-TERM	**Reset.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	EXEC-TERM
87	EXEC-TERM	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	EXEC-TERM
88	EXEC-TERM	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	EXEC-TERM
89	EXEC-TERM	**DP-slave Stopped.ind (AREP)** =>	EXEC-TERM
90	RESETTING	**DeleteInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) => AREP := GetAREP () Reset.rsp(+) (AREP)	IDLE

#	Current State	Event /Condition =>Action	Next State
91	RESETTING	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	RESETTING
92	RESETTING	**Stop.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	RESETTING
93	RESETTING	**Resume.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	RESETTING
94	RESETTING	**Reset.ind (AREP, Slot Number, FI Index)** /(CheckAREP(AREP) = FALSE) => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	RESETTING
95	RESETTING	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	RESETTING
96	RESETTING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = TRUE) =>	WAIT-DEL-IN-USE-2
97	RESETTING	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** /(CheckAREP(AREP) = FALSE) =>	RESETTING
98	RESETTING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = TRUE) =>	WAIT-DEL-IN-USE-2
99	RESETTING	**DP-slave Stopped.ind (AREP)** /(CheckAREP(AREP) = FALSE) =>	RESETTING

#	Current State	Event /Condition =>Action	Next State
100	WAIT-DEL-IN-USE-2	**DeleteInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	IDLE
101	WAIT-DEL-IN-USE-2	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	WAIT-DEL-IN-USE-2
102	WAIT-DEL-IN-USE-2	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	WAIT-DEL-IN-USE-2
103	WAIT-DEL-IN-USE-2	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	WAIT-DEL-IN-USE-2
104	WAIT-DEL-IN-USE-2	**Reset.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	WAIT-DEL-IN-USE-2
105	WAIT-DEL-IN-USE-2	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	WAIT-DEL-IN-USE-2
106	WAIT-DEL-IN-USE-2	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	WAIT-DEL-IN-USE-2
107	WAIT-DEL-IN-USE-2	**DP-slave Stopped.ind (AREP)** =>	WAIT-DEL-IN-USE-2
108	WAIT-SET-IN-USE	**SetInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	RUNNING
109	WAIT-SET-IN-USE	**SetInUse.cnf (-) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	IDLE

#	Current State	Event /Condition =>Action	Next State
110	WAIT-SET-IN-USE	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	WAIT-SET-IN-USE
111	WAIT-SET-IN-USE	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	WAIT-SET-IN-USE
112	WAIT-SET-IN-USE	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	WAIT-SET-IN-USE
113	WAIT-SET-IN-USE	**Reset.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	WAIT-SET-IN-USE
114	WAIT-SET-IN-USE	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	WAIT-SET-IN-USE
115	WAIT-SET-IN-USE	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	WAIT-SET-IN-USE
116	WAIT-SET-IN-USE	**DP-slave Stopped.ind (AREP)** =>	WAIT-SET-IN-USE
117	WAIT-DEL-IN-USE	**DeleteInUse.cnf (+) (FI Identifier, LR Identifier)** /(CheckLRIdentifier (LR Identifier) = TRUE) =>	STOPPED
118	WAIT-DEL-IN-USE	**Start.ind (AREP, Slot Number, FI Index, Execution Argument)** => Error Code := state conflict Start.rsp(-) (AREP, Error Code)	WAIT-DEL-IN-USE
119	WAIT-DEL-IN-USE	**Stop.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Stop.rsp(-) (AREP, Error Code)	WAIT-DEL-IN-USE

#	Current State	Event /Condition =>Action	Next State
120	WAIT-DEL-IN-USE	**Resume.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Resume.rsp(-) (AREP, Error Code)	WAIT-DEL-IN-USE
121	WAIT-DEL-IN-USE	**Reset.ind (AREP, Slot Number, FI Index)** => Error Code := state conflict Reset.rsp (-) (AREP, Error Code)	WAIT-DEL-IN-USE
122	WAIT-DEL-IN-USE	**Get FI State.ind (AREP, Slot Number, FI Index)** => FI State := GetFIState() Get FI State.rsp (+) (AREP, FI State)	WAIT-DEL-IN-USE
123	WAIT-DEL-IN-USE	**Abort.ind (AREP, Locally Generated, Subnet, Instance, Reason Code, Additional Detail)** =>	WAIT-DEL-IN-USE
124	WAIT-DEL-IN-USE	**DP-slave Stopped.ind (AREP)** =>	WAIT-DEL-IN-USE

6.2.9 Time ASE

6.2.9.1 Overview

The Time ASE specifies the Clock Synchronization mechanism to synchronies the time used within devices of a network. This time could be used for example to add a time value to an alert information and to sequence messages in a time wise order.

A system consists of one or more Time Master and several Time Receiver. Clock synchronization is done by the communication system. Adjustment and management of the clock is task of the application.

Time Master tasks are:

- Sending Time Event and Clock Value within Clock Sync Interval
- Resolution of Time Master conflicts

Report Errors in Clock Synchronisation

Time Receiver tasks are:

- Monitoring and check of the clock via clock sync interval
- Control of clock synchronization sequence
- Interpretation clock value
- Check status in clock value

Adjustment of local time (not part of the Clock Synchronisation specification)

The time object specifies two classes:

- Slave Time Class

Link Time Class

The Slave Time Class comprises the Time Receiver tasks. The Link Time Class comprises Time Master and Time Receiver tasks. But Link Time Class differ from Slave Time Class definition.

The difference is based on a system architecture which provide one or more Time Master in a hierarchical topology and Time Receiver (DP-slaves) which receive a synchronized time value to set a clock within the device, see

Figure 28.

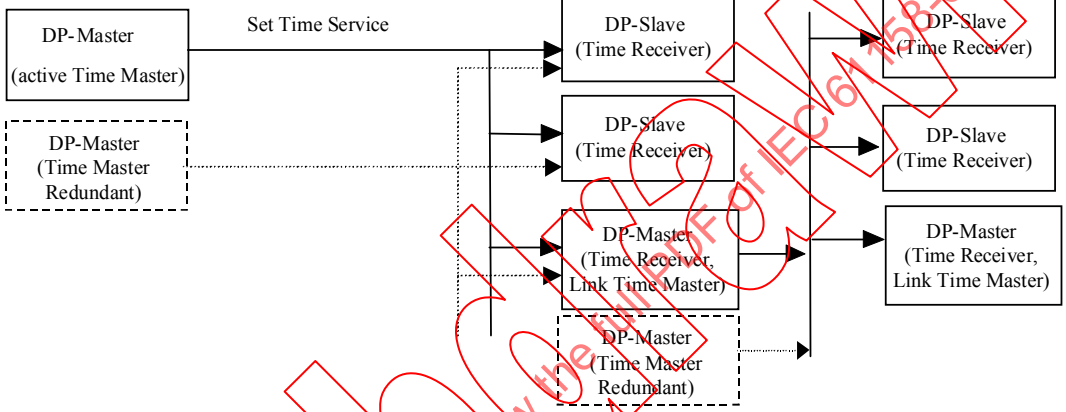

Figure 28 – System architecture

Two services are defined:

— Set Time

— Sync Interval Violation

The Set Time service is requested by the DP-master device and the Sync Interval Violation service through DP-slave devices.

Synchronization of time between devices on a fieldbus segment and a Time Master application is provided in parallel with other communication functions. The DP-master (Class 1) operates as the active and if required as the redundant Time Master. If a DP-slave supports the Clock Synchronization mechanism the device attribute DPV1 has to be set to TRUE.

The scheme used is a "backwards time based correction". This results in very few real-time constraints being imposed on a field device. There is no requirement for generating messages at precise instants. Instead, knowledge of when a special Timer event message has been broadcasted is subsequently distributed and used to calculate appropriate clock adjustments.

NOTE The clock synchronization assumes that there are no delays between the invocation of the Set Time service and the begin of the processing of the related service in the data-link layer.

The clock synchronization protocol operates in parallel with the other AL functions. The clock synchronization interval specifies the rate at which the Time Master initiates the clock synchronization sequence. The value of this parameter is a system design issue. It shall be the same for all devices on the network.

The value of the clock synchronization interval is a system design issue. For each station, there is a minimum required clock synchronization interval given by the desired clock accuracy divided by the clock drift rate. The value for the system should be set to the minimum required value for all stations on the network segment. The value should not be set so large that it causes intolerable start-up delays when devices, such as hand-held terminals, are added to a network.

6.2.9.2 Time class specification

6.2.9.2.1 Slave time class specification

6.2.9.2.1.1 Template

DP ASE:			**Time ASE**
CLASS:		**Slave Time**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Time Data Description
3.	(o)	Attribute:	Local Time Diff
4.	(o)	Attribute:	Time Last Rcvd
5.	(o)	Attribute:	Last Calc Diff
6.	(m)	Attribute:	CS Status
7.	(o)	Attribute:	Summertime
SERVICES:			
1.	(m)	OpsService:	Set Time
2.	(m)	OpsService:	Sync Interval Violation

6.2.9.2.1.2 Attributes

Implicit
The attribute Implicit indicates that the Slave Time object is implicitly addressed by the services.

Time Data Description
This attribute contains a Simple Data Description. Only the Data Type Network Time shall be used. For Data Description the template given in IEC 61158-5-10, 5.1 is used.

Local Time Diff
This optional attribute contains the value that needs to be added to the value of the Slave Time object to obtain the local time.

The default value shall be 0 (no local time used).

Attribute Type: Network Time Difference

Time Last Rcvd
This optional attribute contains the parameter Time Value of the last Set Time service.

Attribute Type: Network Time

Time Master Addr
This optional attribute contains the address of the device from whom the previous Set Time service was received.

Attribute Type: Unsigned8

Allowed Value: 0 to 126

Last Calc Diff
This optional attribute contains the difference between the value of the Slave Time object and the parameter Time Value of the Set Time service plus Delay Factor at the receipt of the last Set Time service.

Attribute Type: Network Time Difference

CS Status
This Boolean attribute contains the current status of the synchronization.

Attribute Type: Boolean

The allowed values are shown in Table 189.

Table 189 – CS status

Value	Meaning
FALSE	Synchronized
TRUE	Not Synchronized Time since last reception of a Set Time service is too long (greater than 3 Clock Sync Intervals).

Summertime
This optional Boolean attribute indicates whether the value of the attribute Local Time Diff is based on Summertime or Wintertime.

Attribute Type: Boolean

The allowed values are shown in Table 190.

Table 190 – Summertime

Value	Meaning
FALSE	Wintertime
TRUE	Summertime

6.2.9.2.1.3 Invocation of the slave time object

For the invocation of the Slave Time object the following rules apply:

— Only one Slave Time object shall be invoked in a DP-slave.

6.2.9.2.2 Link time class specification

6.2.9.2.2.1 Template

DP ASE: **Time ASE**
CLASS: **Link Time**
CLASS ID: not used
PARENT CLASS: TOP
ATTRIBUTES:

1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	Time Data Description
3.	(o)	Attribute:	Local Time Diff
4.	(o)	Attribute:	Time Last Rcvd
5.	(o)	Attribute:	Last Calc Diff
6.	(m)	Attribute:	Synchronization Active
7.	(o)	Attribute:	Announcement Hour
8.	(o)	Attribute:	Summertime
9.	(o)	Attribute:	Accuracy

SERVICES:

1.	(m)	OpsService:	Set Time
2.	(m)	OpsService:	Sync Interval Violation

6.2.9.2.2.2 Attributes

Implicit
The attribute Implicit indicates that the Link Time object is implicitly addressed by the services.

Time Data Description
This attribute contains a Simple Data Description. Only the Data Type Network Time shall be used. For Data Description the template given in IEC 61158-5-10, 5.1 is used.

Local Time Diff
This optional attribute contains the value that needs to be added to the value of the Link Time object to obtain the local time. It shall be a multiple of 30 minutes, between 0 and 24 hours.

The default value shall be 0 (no local time used).

Attribute Type: Network Time Difference

Time Last Rcvd
This optional attribute contains the parameter Time Value of the last Set Time service.

Attribute Type: Network Time

Last Calc Diff
This optional attribute contains the difference between the value of the Link Time object and the parameter Time Value of the Set Time service plus Delay Factor at the receipt of the last Set Time service.

Attribute Type: Network Time Difference

Synchronization Active
This Boolean attribute indicates whether the value of the Link Time object is synchronized or not. This attribute is set to 0 if either this station has received a Set Time service with the equivalent Parameter set to 0 or if the time since last reception of a Set Time service is too long (greater than 3 Clock Sync Intervals).

Attribute Type: Boolean

The allowed values are shown in Table 191.

Table 191 – Synchronization active

Value	Meaning
FALSE	Synchronization Failed
TRUE	Synchronization Active

Default value: FALSE

Announcement Hour
This optional Boolean attribute indicates whether a change from Summertime to Wintertime or vice versa shall be performed with the next change of the hour.

Attribute Type: Boolean

The allowed values are shown in Table 192.

Table 192 – Announcement hour

Value	Meaning
FALSE	No change with the next hour change.
TRUE	Change from Summertime to Wintertime or vice versa with the next hour change.

Summertime
This optional Boolean attribute indicates whether the value of the attribute Local Time Diff is based on Summertime or Wintertime.

Attribute Type: Boolean

The allowed values are shown in Table 193.

Table 193 – Summertime

Value	Meaning
FALSE	Wintertime
TRUE	Summertime

Accuracy
This optional attribute indicates the precision of the value of the Link Time object, i.e. which bit positions are relevant. The value will be taken from the Set Time service parameter Accuracy.

Attribute Type: Unsigned8

The allowed values are shown in Table 194.

Table 194 – Accuracy

Value	Meaning
0	< 1 ms
1	< 10 ms
2	< 100 ms
3	< 1 s

Default value: 0

6.2.9.2.2.3 Invocation of the link time object

For the invocation of the Link Time object the following rules apply:

— Only one Link Time object shall be invoked in a Link.

6.2.9.3 Time service specification

6.2.9.3.1 Set time

This confirmed service shall be used to set the value and some attributes of a Slave Time or Link Time object. This service shall be used only in conjunction with the MS3 AR. This AR provides the facility to address a group of or all present devices on a fieldbus segment (multicast/broadcast) and therefore the corresponding Slave Time or Link Time objects will be set simultaneously. Due to this principle no confirmation of reception can be provided at the requesting side.

Table 195 shows the parameters of the Set Time service.

Table 195 – Set time

Parameter name	Req	Ind	Cnf
Argument	M	M(=)	
AREP	M	M	
Time Value	M	M	
Local Time Diff	M	M(=)	
Summertime	M	M(=)	
Accuracy	M	U(=)	
Synchronization Active	M	U(=)	
Announcement Hour	M	U(=)	
Result(+)			S
AREP			M
Result(-)			S
AREP			M
Status			M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

Time Value
This Parameter contains the value for the Link Time or Slave Time object.

Local Time Diff
This attribute contains the value for the attribute Local Time Diff of the Link Time or Slave Time object.

Summertime
This attribute contains the value for the attribute Summertime of the Link Time or Slave Time object.

Accuracy
This attribute contains the value for the attribute Accuracy of the Link Time object and shall be present only in the indication if a Link Time object is present.

Synchronization Active
This attribute contains the value for the attribute Synchronization Active of the Link Time object and shall be present only in the indication if a Link Time object is present.

Announcement Hour
This attribute contains the value of the attribute Announcement Hour of the Link Time object and shall be present only in the indication if a Link Time object is present.

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

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

Status:
This parameter provides detailed information about the occurred error.

Allowed values: LS, LR, DS, SV, IV

6.2.9.3.2 Sync interval violation

The Sync Interval Violation service signals to the application of the Time Receiver (DP-slave) that no Set Time service has been received during the last 2 Clock Sync Intervals. This service shall be used only in conjunction with the MS3 AR.

Table 196 shows the parameter of the Sync Interval Violation service.

Table 196 – Sync interval violation

Parameter name	Ind
Argument	M
AREP	M

Argument
The argument shall convey the service specific parameters of the service request.

AREP
This parameter is the local identifier for the desired AR.

6.2.10 AR ASE

6.2.10.1 Overview

6.2.10.1.1 General

This ASE specifies the structure of the ARs including the necessary DL parameters for the specific CRs. The AR ASE in each device manages an end point of the AR and all related CRs, and maintains their local context.

An AR is characterized by all of the following basic roles of the AREP used in fieldbus DP:

AR cardinality

— one-to-one

In a one-to-one AR an AP communicates with exactly one remote AP.

— one-to-many

In a one-to-many AR the AP of a DP-master (Class 2) communicates with the APs of a group of DP-masters (Class 1). This comprises the simultaneous communication to the defined group of DP-masters (Class 1). Exactly one AP is addressed per device.

— many-to-many

In a many-to-many AR the APs of a DP-master (Class 1) and optional the APs of one or several DP-master (Class 2) and optional the Default-APs of one or several DP-slave communicate with the Default-APs of a group of DP-slaves. This comprises the sequential as well as the simultaneous communication to the defined group of DP-slaves. Exactly one AP is addressed per device.

Device type relations:

Master-slave (MS) communication
between DP-master (Class 1) and DP-slaves

between DP-master (Class 2) and DP-slaves

Master-master (MM) communication
between DP-master (Class 2) and DP-master (Class 1)

6.2.10.1.2 Communication relationship

6.2.10.1.2.1 General

An AR is mapped to one or more communication relationships (CRs). A CR carries out the exchange of information based on a set of rules.

6.2.10.1.2.2 Communication relationship endpoint

CRs are defined as a set of co-operating Communication Relationship End Points (CREPs). The local context of an CR endpoint is used to control the conveyance of APDUs on the CR. Figure 29 illustrates the Assignment of Communication Relationship to Application Relationship.

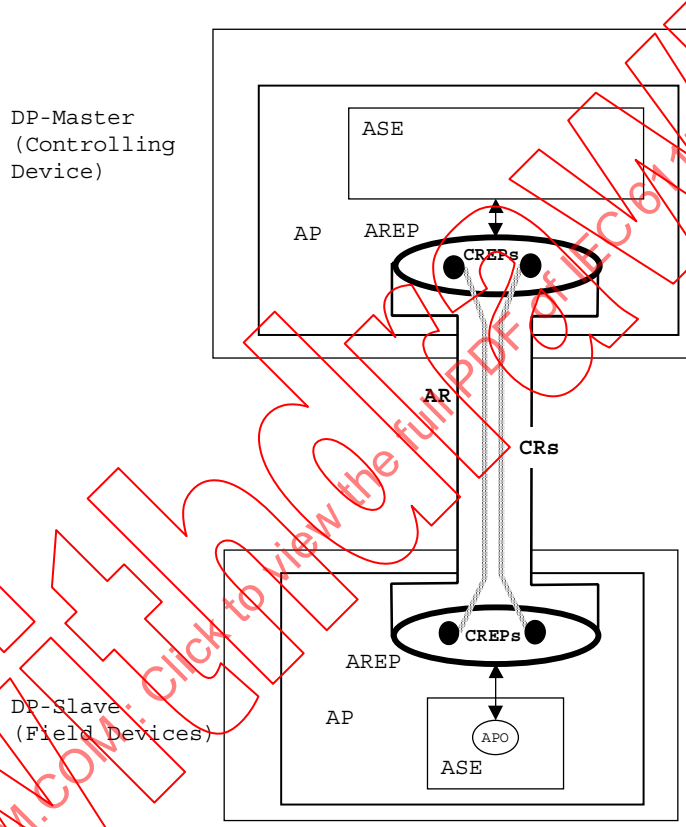

Figure 29 – Assignment of communication relationship to application relationship

6.2.10.1.2.3 Buffer and queues

CREPs may be modeled as a queue or as a buffer. Application data transferred over a queued CREP are delivered in the order received for conveyance. The transfer of application data over a buffered CREP is different. In this case, application data is placed in an application layer buffer for transfer.

When the CREP receives another conveyance request, it replaces the previous contents of the buffer whether or not transmitted. Once application data are written into a buffer for transfer, it is preserved in the buffer until the next application data to be transmitted replaces it. While in the buffer, application data may be read more than once without deleting it from the buffer or changing its contents.

At the receiving end, the operation is similar. The receiving application layer places the received application data into a buffer for access by the different ASEs. When subsequent

application data are received, it overwrites the previous application data in the buffer whether or not it was read by the corresponding ASE. Reading the application data from the buffer is not destructive it does not destroy or change the contents of the buffer, allowing the contents to be read from the buffer one or more times.

6.2.10.1.2.4 Communication relationship endpoint roles

An communication relationship is characterized by all of the following basic roles of the CREP used in fieldbus DP:

CR cardinality

— one-to-one:

In a one-to-one communication relationship a CREP communicates with exactly one remote CREP.

— one-to-many:

In a one-to-many relationship one CREP communicates simultaneously with the CREPs of a group of devices. A one-to-many communication relationship does not permit confirmation of the execution of a request.

Connection mode:

connection-oriented

In a connection-oriented application relationship a logical connection is established between application processes. Three different phases are identified in connection-oriented ARs:

— connection establishment phase

— data transfer phase and

— connection release phase

Exchanging data on a connection shall only be possible after a successful connection establishment. In the connection establishment phase a request to establish the connection is announced at the remote application process (e.g. Initiate service). The request to establish the connection-oriented AR contains the information about the context to be used in the data transfer phase. The remote application informs the requesting application process about the acceptance of the context. If the context is accepted both application processes are in the data transfer phase and are able to communicate with each other according to the agreed context. Connection-oriented AR can be monitored. A connection-oriented AR is dissolved by its release (e.g. Abort service).

connectionless

Connectionless application relationships have neither a connection establishment nor a connection release. They are always in the data transfer phase. It is not possible to monitor the connectionless AR.

Transmission types:

cyclic

The transmission type "cyclic" means that after an initial request by the application process the application relationship is established and afterwards the data transfer is executed periodically.

acyclic

The transmission type "acyclic" means that a single transfer is executed after each request by the application process.

both

The transmission type "both" means that a cyclic as well as acyclic transfer is supported.

Transmission characteristics

Buffer to Buffer

The transmission is characterized by a buffer to buffer transfer.

Queue to Queue
The transmission is characterized by a queue to queue transfer.

Buffer to Queue
The transmission is characterized by a buffer to queue transfer.

Queue to Buffer
The transmission is characterized by a queue to buffer transfer.

Service types
Confirmed
Only confirmed services are allowed.

Unconfirmed
Only unconfirmed services are allowed.

Transmission directions:
Bi-directional
APDUs are transmitted in both direction.

Unidirectional
APDUs are transmitted only in one direction.

Service relations:
Client-Server communication
The services are used according to the Client/Server model.

Publisher-Subscriber communication
The services are used according to the Publisher/Subscriber model.

6.2.10.1.2.5 Overview of communication relationships

The DP Application Layer uses the following types of communication relationships:

BBUB (Buffer Buffer Unconfirmed Bidirectional)

This type of communication relationship is characterized by the following role attributes:

— one to one
— connection-oriented
— cyclic
— buffer to buffer
— unconfirmed
— bi-directional
— Client/Server

BBUU-OM (Buffer Buffer Unconfirmed Unidirectional - One to Many)

This type of communication relationship is characterized by the following role attributes:

— one to many
— connection-oriented
— cyclic
— buffer to buffer
— unconfirmed
— unidirectional
— Publisher/Subscriber

BQUU (Buffer Queue Unconfirmed Unidirectional)

This type of communication relationship is characterized by the following role attributes:

— one to one
— connectionless
— acyclic
— buffer to queue
— unconfirmed
— unidirectional
— Client/Server

QQUU-OM (Queue Queue Unconfirmed Unidirectional - One to Many)

This type of communication relationship is characterized by the following role attributes:

— one to many
— connectionless
— acyclic
— queue to queue
— unconfirmed
— unidirectional
— Client/Server

BBUU (Buffer Buffer Unconfirmed Unidirectional)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connectionless
- acyclic
- buffer to buffer
- unconfirmed
- unidirectional

Client/Server

QQCB-CO (Queue Queue Confirmed Bi-directional - Connection-Oriented)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connection-oriented
- acyclic
- queue to queue
- confirmed
- bidirectional

Client/Server

QQUU-OO (Queue Queue Unconfirmed Unidirectional - One to One)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connectionless
- acyclic
- queue to queue
- unconfirmed
- unidirectional

Client/Server

QBCB (Queue Buffer Confirmed Bi-directional)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connectionless
- acyclic
- queue to buffer
- confirmed
- bi-directional

Client/Server

QQCB-CL (Queue Queue Confirmed Bi-directional - Connectionless)

This type of communication relationship is characterized by the following role attributes:

- one to one
- connectionless
- acyclic
- queue to queue
- confirmed
- bi-directional

Client/Server

6.2.10.2 Fieldbus ARs

6.2.10.2.1 MS0 application relationship

The MS0 AR reflects the application relationship (see Figure 30) between the application processes of one DP-master (Class 1) and optional the APs of one or several DP-master (Class 2) with the Default AP of all related DP-slaves, additionally the MS0 AR provides the application relationship between the Default-APs of one or several DP-slaves with the Default-APs of a group of DP-slaves for the following purposes:

- cyclic exchange of the I/O data (DP-master (Class 1)) (Client/Server)
- cyclic exchange of the Input Data (DP-slave) (Publisher/Subscriber)
- acyclic data transfer for parameterization, configuration and diagnosis (DP-master (Class 1))
- acyclic transfer of commands to a set of field devices (DP-master (Class 1))
- cyclic transfer of synchronization messages to a set of field devices (DP-master (Class 1))
- acyclic read of the I/O data (DP-master (Class 2))
- acyclic read of configuration information (DP-master (Class 2))
- acyclic read of diagnosis information (DP-master (Class 2))

acyclic write of remanent parameter (DP-master (Class 2))

This type of application relationship is characterized by the following role attributes:

- many to many

Master-slave communication

The MS0 AR contains the following CR types:

- BBUB (Data Exchange)
- BBUU-OM (DXB)
- BQUU (Prm, Ext Prm and Cfg)
- BBUU (Slave Diag)
- QQUU-OM (Global Control)

QBCB (Slave Diag, Read Cfg, Read Input, Read Output, Rem Para)

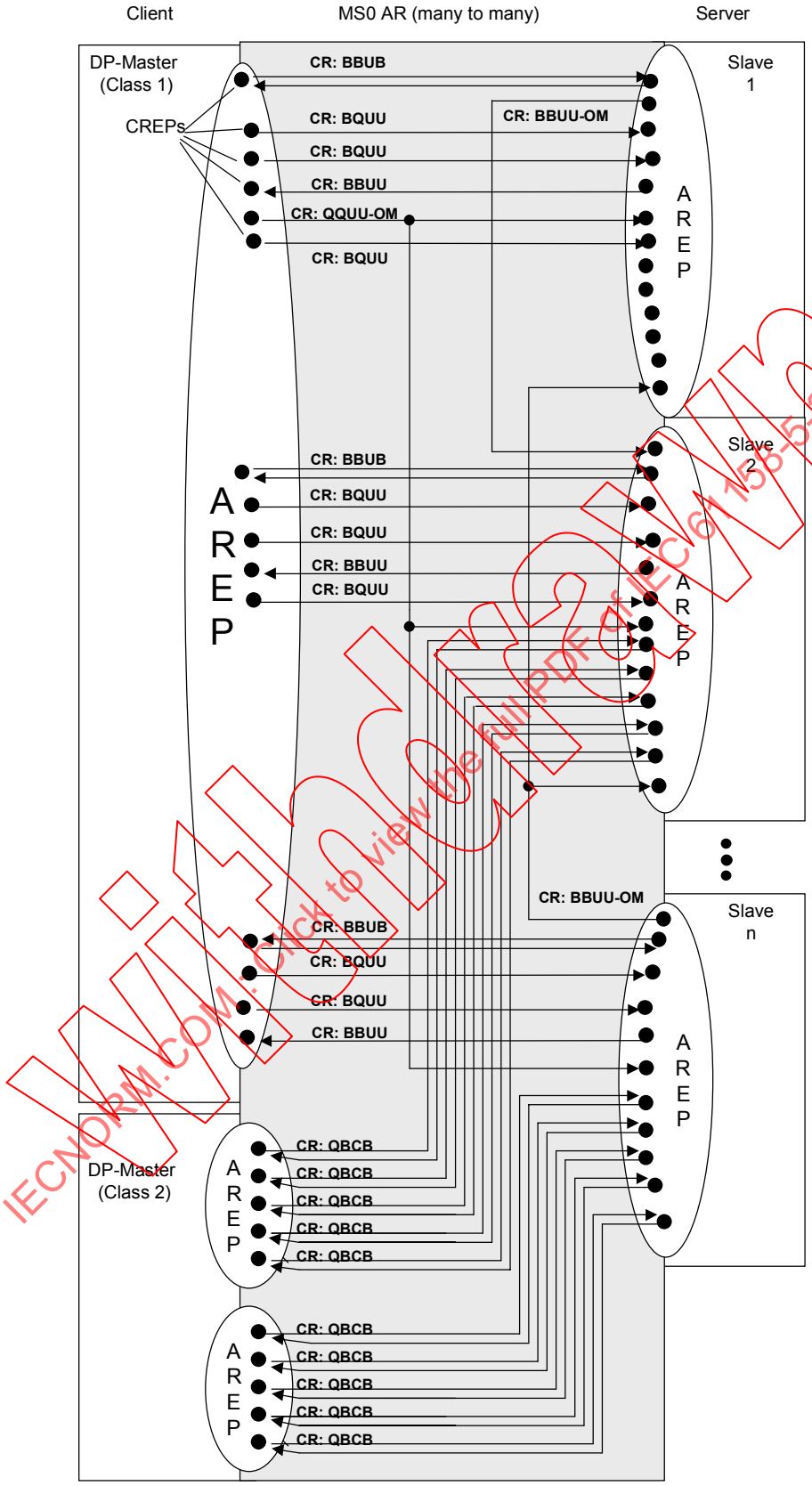

Figure 30 – MS0 application relationship

The DP-Application Layer incorporates buffers to convey the input and output data. These buffers decoupage the communication and data processing. For synchronization purposes two buffers for the Input Data and two buffers for the Outputs are necessary in the DP-Application Layer. If a DP-slave supports no synchronization capabilities only one buffer is needed in each direction. For the data of the subscribed Input Data only one buffer may be provided by the DP-Application Layer. The detailed model is described in the following subclauses. Figure 31 through Figure 34 illustrate the Input and Output Buffer models of a DP-slave.

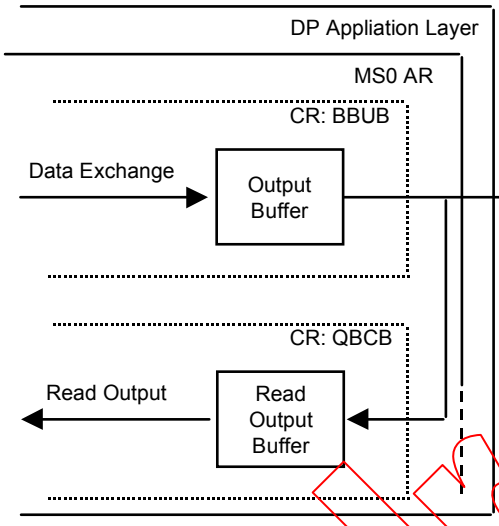

Figure 31 – Output buffer model of a DP-slave without sync functionality

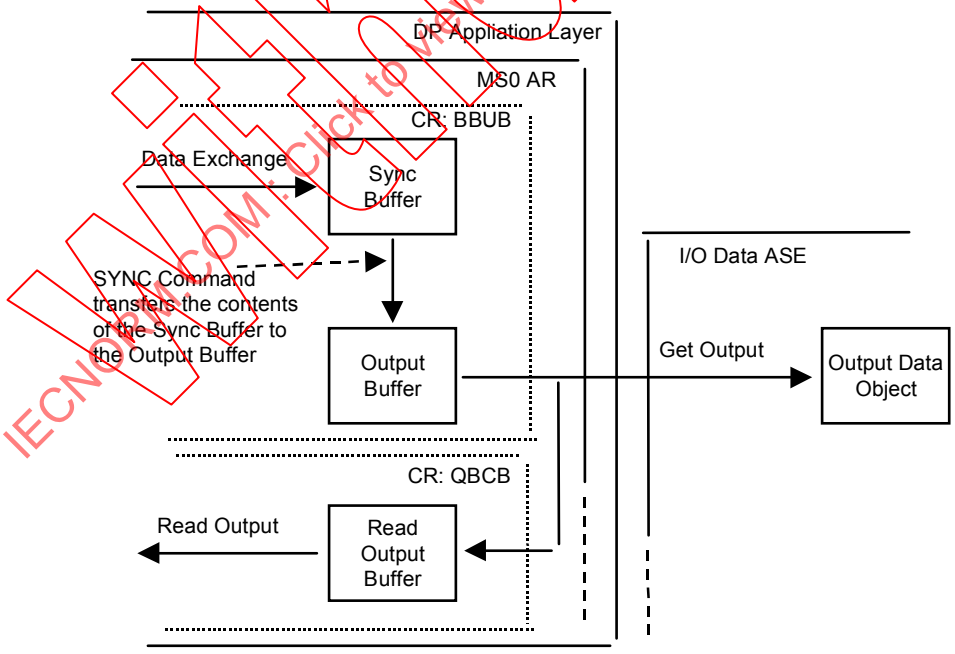

Figure 32 – Output buffer model of a DP-slave with sync functionality

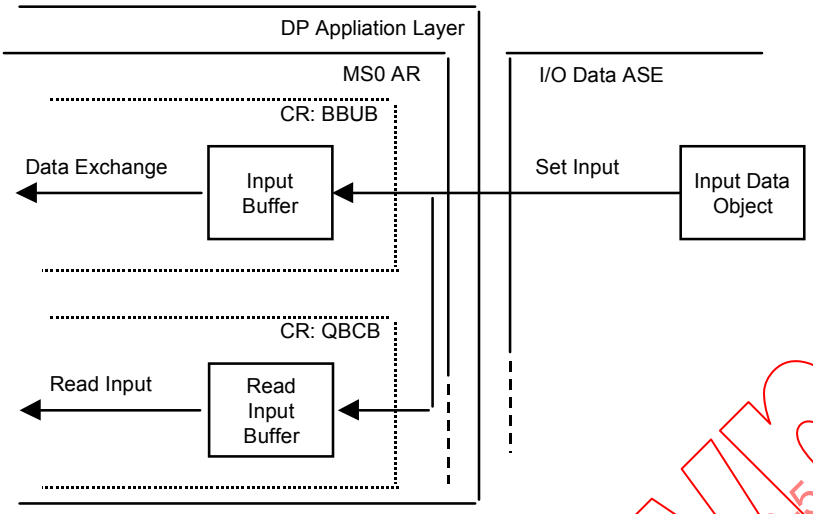

Figure 33 – Input buffer model of a DP-slave without freeze functionality

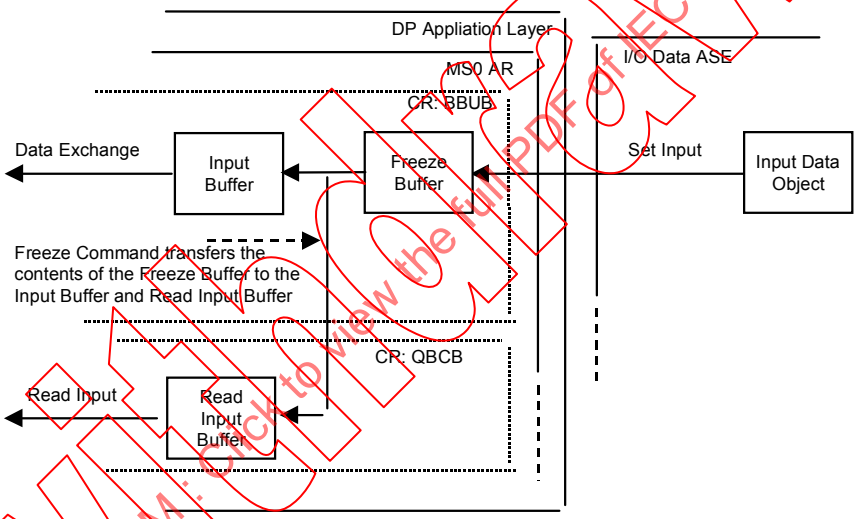

Figure 34 – Input buffer model of a DP-slave with freeze functionality

6.2.10.2.2 MS1 application relationship

The MS1 AR reflects the connection-oriented application relationship (see Figure 35) between the application processes of one DP-master (Class 1) and of one related DP-slave for the following purposes:

- acyclic read and write of variables
- acyclic transfer of alarms
- up- and/or download of LR Data

invocation of stateless and/or state-oriented functions

This type of application relationship is characterized by the following role attributes:

- one to one

Master-slave communication

The MS1 AR contains the following CR types:

- QQCB-CO (Process Data, Alarm Ack,)

QQUU-OO (Alarm)

and optionally the following CR type:

QQCB-CO (Alarm Ack)

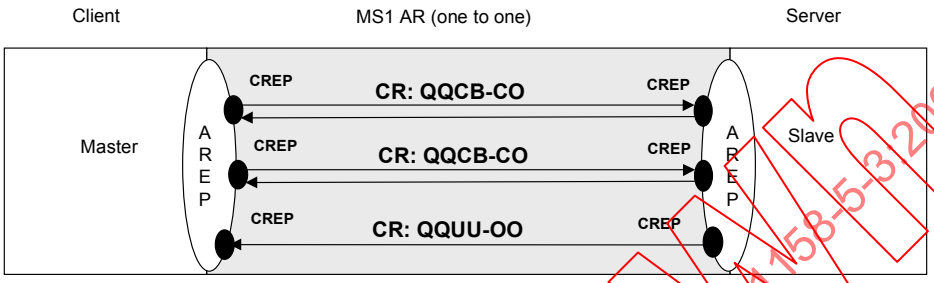

Figure 35 – MS1 application relationship

6.2.10.2.3 MS2 application relationship

6.2.10.2.3.1 General

The MS2 AR reflects the connection-oriented application relationship (see Figure 36) between the application processes of one DP-master (Class 2) and of one related DP-slave for the purpose:

- acyclic read and write of variables
- up- and/or download of LR Data

invocation of stateless and/or state-oriented functions

This type of application relationship is characterized by the following role attributes:

- one to one

Master-slave communication

The MS2 AR contains the following CR type:

- QQCB-CO (Process Data)

QQUU-OO (Resource Manager for Context Management)

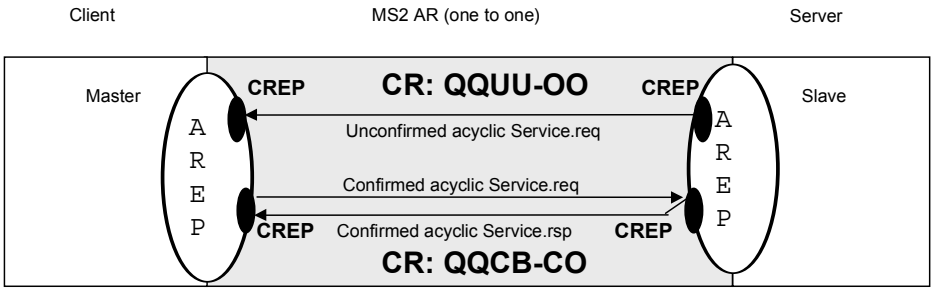

Figure 36 – MS2 application relationship

6.2.10.2.3.2 Communication over multiple interconnected networks

With the Initiate service it is possible to communicate over multiple networks.

The following cases are considered:

- data transfer from one DP-Network to another DP-Network over a link

data transfer from a None-DP-Network to a DP-Network over a gateway

Within such networks each DP-Device including their applications are addressed with the following parameters:

- Network address
- MAC
- API, SCL

The format of these addresses is defined in the Initiate service description. It is necessary to distinguish between DP-Devices which are either used as a link or a communication endpoint. In the case of the link the Initiate-PDU has to be passed to the addressed device. The Network address 0 is reserved for default addressing purposes and therefore shall not be used as a normal network address.

The Figure 37 shows a cut out of an heterogeneous network. Fieldbus DP covers only the marked area.

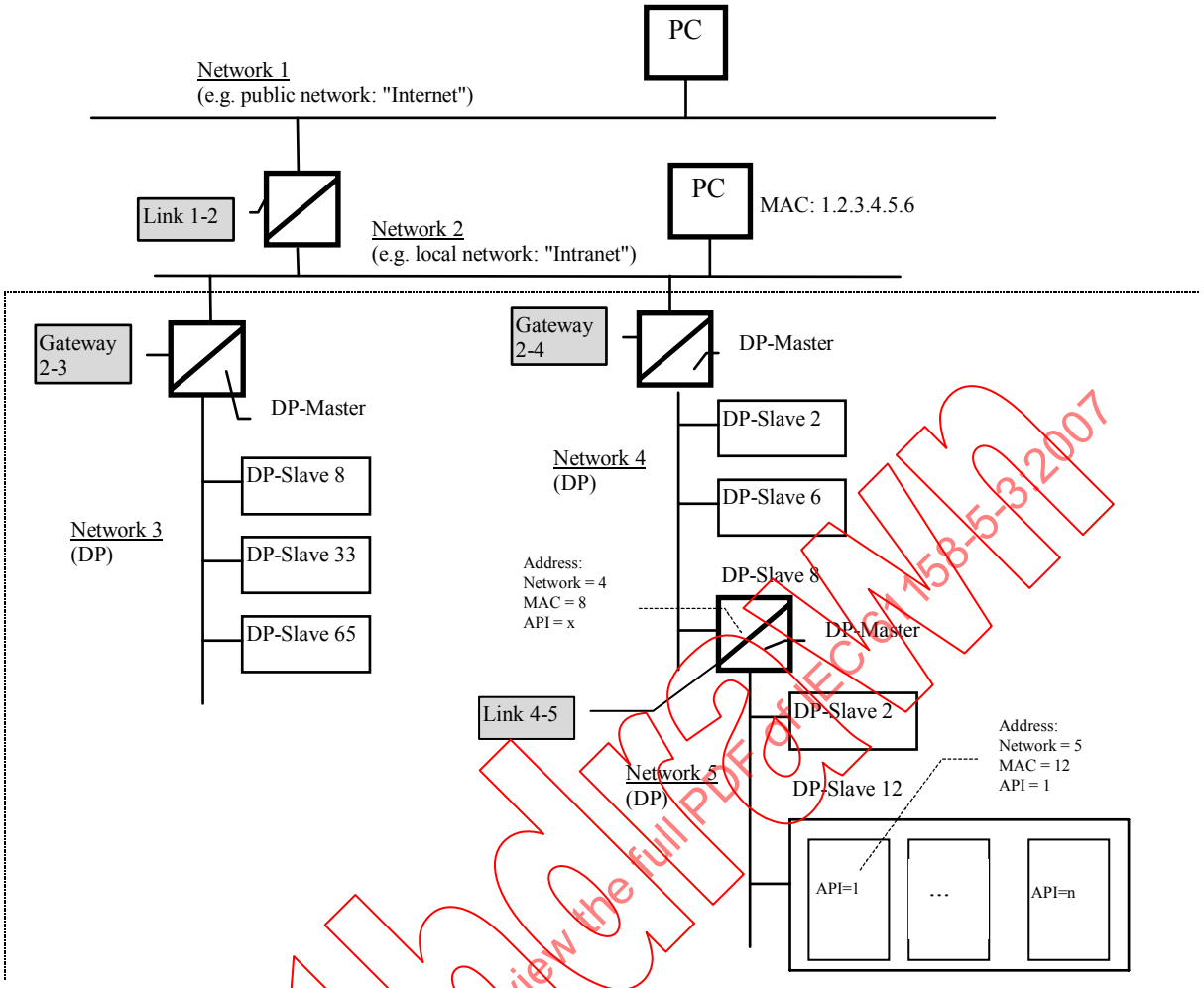

Figure 37 – Example of inter-network communication

The following examples (see Figure 38 through Figure 40) show the exchange of the source- and destination-address parameter of the Initiate.req and the Initiate.res service.

Between the AP 10 of DP-master (Class 2) and AP 1 of the DP-slave a communication relationship will be established. This is done with the Initiate service. After the connection establishment the transfer of data blocks will be realized with the services Read and Write using the parameters Slot Number and Index.

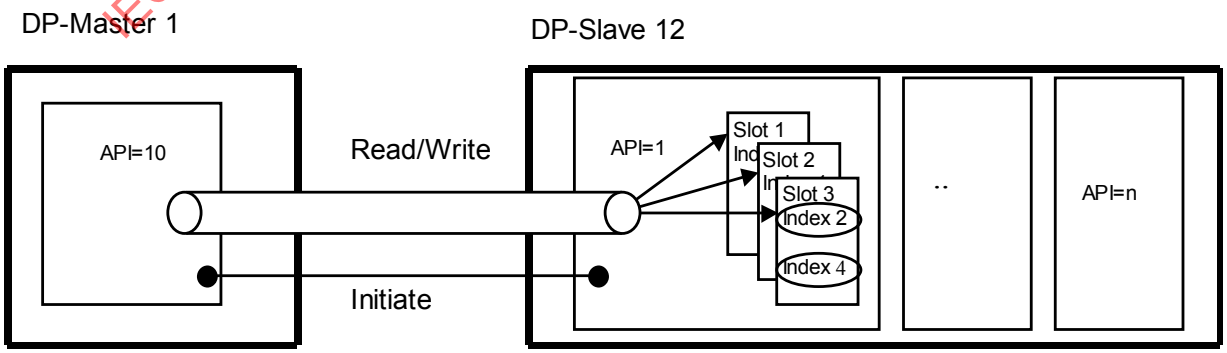

Figure 38 – Example without inter-network addressing

Table 197 shows the parameters of the Initiate service without Inter-Network Addressing.

Table 197 – Parameter of Initiate service without inter-network addressing

Initiate.req (1 -> 12)		Initiate.res (12 -> 1)	
S-Type	0	S-Type	0
S-Len	2	S-Len	2
D-Type	0	D-Type	0
D-Len	2	D-Len	2
S-Addr	API 10, SCL	S-Addr	API 1, SCL
D-Addr	API 1, SCL	D-Addr	API 10, SCL

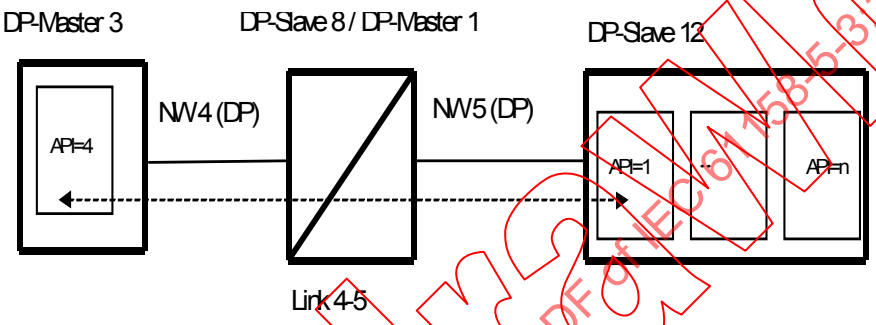

Figure 39 – First example with inter-network addressing

Table 198 shows the parameters of the Initiate service with Inter-Network Addressing.

Table 198 – Parameter of Initiate service with inter-network addressing (first example)

Initiate.req (3 -> 8)		Initiate.res (8 -> 3)	
S-Type	0	S-Type	1
S-Len	2	S-Len	9
D-Type	1	D-Type	0
D-Len	9	D-Len	2
S-Addr	API 4, SCL	S-Addr	API 1, SCL, NW5, 12
D-Addr	API 1, SCL, NW5, 12	D-Addr	API 4, SCL
Initiate.req (1 -> 12)		**Initiate.res (12 -> 1)**	
S-Type	1	S-Type	0
S-Len	9	S-Len	2
D-Type	0	D-Type	1
D-Len	2	D-Len	9
S-Addr	API 4, SCL, NW4, 3	S-Addr	API 1, SCL
D-Addr	API 1, SCL	D-Addr	API 4, SCL, NW4, 3

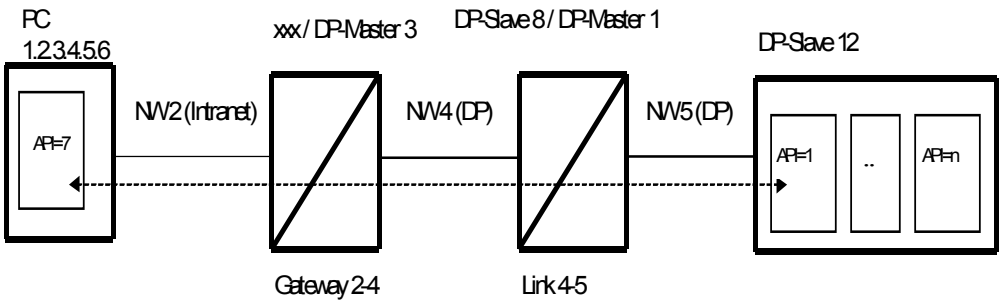

Figure 40 – Second example with inter-network addressing

Table 199 shows the parameters of the Initiate service with Inter-Network Addressing.

Table 199 – Parameter of Initiate service with inter-network addressing (second example)

Initiate-REQ-PDU (3 -> 8)		**Initiate-RES-PDU (8 -> 3)**	
S-Type	1	S-Type	1
S-Len	14	S-Len	9
D-Type	1	D-Type	1
D-Len	9	D-Len	14
S-Addr	API 7, SCL, NW2, 1.2.3.4.5.6	S-Addr	API 1, SCL, NW5, 12
D-Addr	API 1, SCL, NW5, 12	D-Addr	API 7, SCL, NW2, 1.2.3.4.5.6
Initiate-REQ-PDU (1 -> 12)		**Initiate-RES-PDU (12 -> 1)**	
S-Type	1	S-Type	0
S-Len	14	S-Len	2
D-Type	0	D-Type	1
D-Len	2	D-Len	14
S-Addr	API 7, SCL, NW2, 1.2.3.4.5.6	S-Addr	API 1, SCL
D-Addr	API 1, SCL	D-Addr	API 7, SCL, NW2, 1.2.3.4.5.6

6.2.10.2.4 MS3 application relationship

The MS3 AR reflects the connectionless relationship (see Figure 41) between the application processes of a DP-master and of a set of related DP-slaves for the purpose:

— synchronization of time

This type of application relationship is characterized by the following role attributes:

— one to many

— Master-slave communication

The MS3 AR contains the following CR type:

— QQUU-OM (Time)

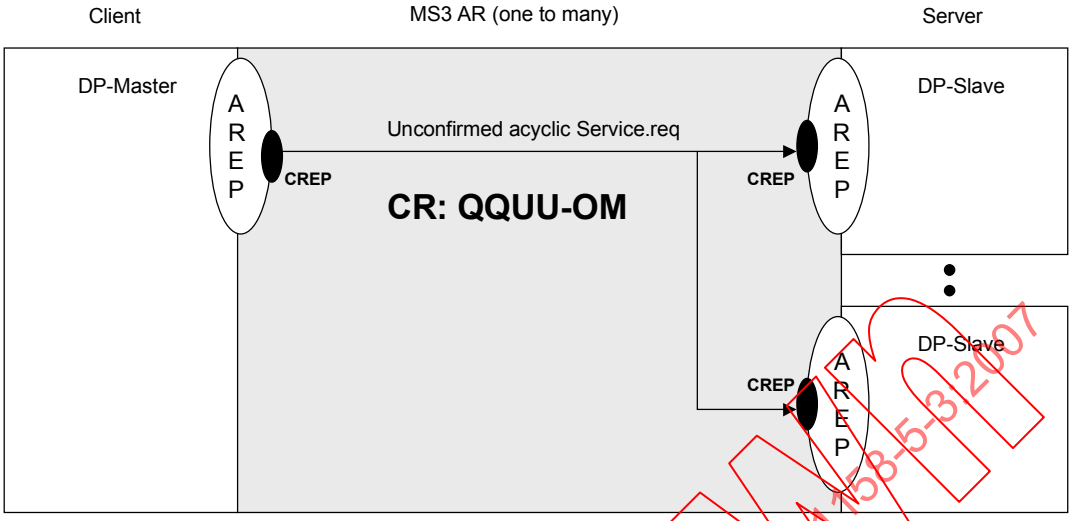

Figure 41 – MS3 application relationship

6.2.10.2.5 MM1 application relationship

The MM1 AR reflects the connectionless relationship (see Figure 42) between the application processes of one configuration device (Master Class 2) and of one related controlling device (Master Class 1) for the following purposes:

- upload and download for configuration information
- upload of diagnosis information

activation of the previously transferred configuration

This type of application relationship is characterized by the following role attributes:

- one to one

Master-master communication

The MM1 AR contains the following CR type:

QQCB-CL (MM1(Master Diag/Start Seq/End Seq/Download/Upload/Act Param))

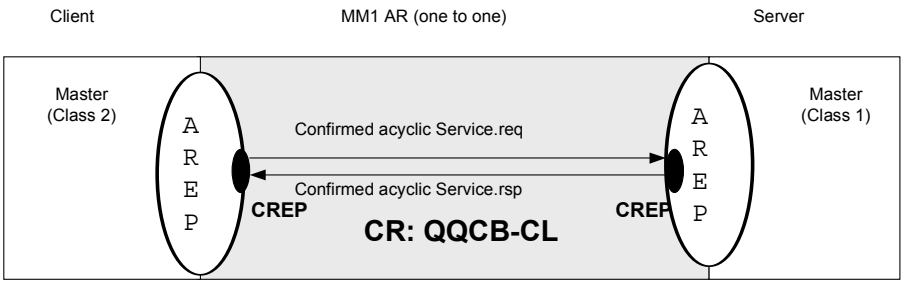

Figure 42 – MM1 application relationship

6.2.10.2.6 MM2 application relationship

The MM2 AR reflects the connectionless relationship (see Figure 43) between the application processes of one configuration device (Master Class 2) and of a set of related controlling devices (Master Class 1) for the purpose:

activation of the previously transferred configurations

This type of application relationship is characterized by the following role attributes:

- one to many

Master-master communication

The MM2 AR contains the following CR type:

QQUU-OM (Act Para Brct)

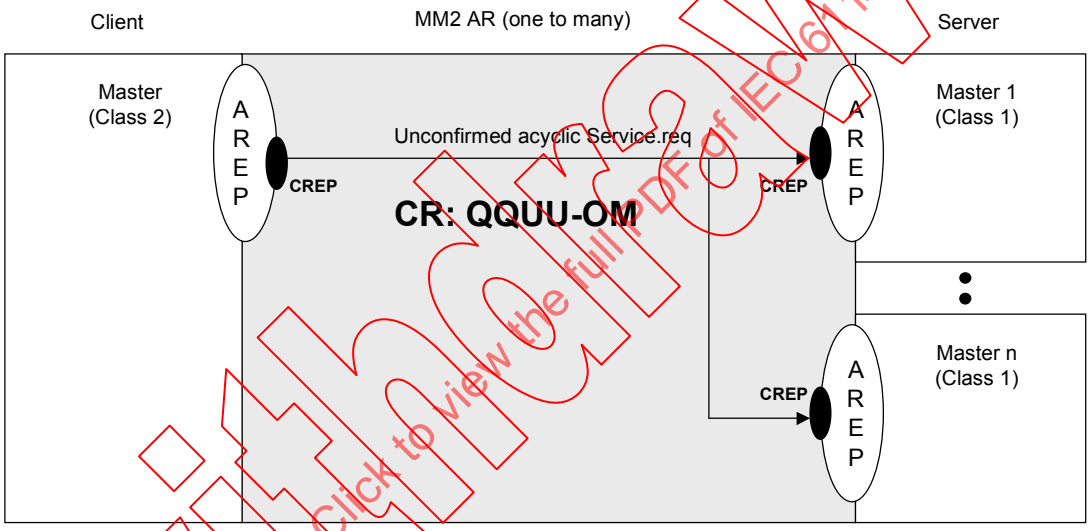

Figure 43 – MM2 application relationship

6.2.10.3 Application relationship list class specification

The ARL Class specification is distinguished between DP-slave, DP-master (Class 1) and DP-master (Class 2).

A static attribute is assigned by a load service and are not changed during operation, whereas a dynamic attribute may set by the load services and may be changed during operation.

6.2.10.3.1 ARL DP-slave class specification

6.2.10.3.1.1 Template

The ARL DP-slave object is described by the following template:

DP ASE:			**AR ASE**
CLASS:	**ARL DP-slave**		
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(c)	Constraint:	at least one entry with AR Type = MS2
2.1.	(m)	Attribute:	Min Send Timeout

2.2.	(m) Attribute:	List of Resource Manager Entries
2.2.1.	(m) Attribute:	API
2.2.2.	(m) Attribute:	SCL
2.2.3.	(m) Attribute:	Max Len Data Unit
2.2.4.	(m) Attribute:	Send Timeout
2.2.5.	(m) Attribute:	AREP
2.2.6.	(m) Attribute:	S_SAP_index
2.2.7.	(m) Attribute:	In Use
2.3.	(m) Attribute:	List of S_SAP_indices
2.3.1.	(m) Attribute:	S_SAP_index
3.	(m) Attribute:	List of ARL Entries
3.1.	(m) Attribute:	AREP
3.2.	(m) Attribute:	AR Type
3.3.	(m) Attribute:	List of related CRL Entries
3.3.1.	(m) Attribute:	CREP
3.4.	(c) Constraint:	AR Type = MS0
3.4.1.	(m) Attribute:	Prm Req
3.4.2.	(m) Attribute:	Prm Fault
3.4.3.	(m) Attribute:	Cfg Fault
3.4.4.	(m) Attribute:	Not Supported
3.4.5.	(m) Attribute:	Station Not Ready
3.4.6.	(m) Attribute:	Stat Diag
3.4.7.	(m) Attribute:	Master Add
3.4.8.	(m) Attribute:	Sync Supported
3.4.9.	(m) Attribute:	Sync Mode
3.4.10.	(m) Attribute:	Freeze Supported
3.4.11.	(m) Attribute:	Freeze Mode
3.4.12.	(m) Attribute:	WD Enabled
3.4.13.	(m) Attribute:	WD Fact 1
3.4.14.	(m) Attribute:	WD Fact 2
3.4.15.	(m) Attribute:	Min TSDR
3.4.16.	(m) Attribute:	Ident Number
3.4.17.	(m) Attribute:	Group Identifier
3.4.18.	(m) Attribute:	DPV1 Supported
3.4.19.	(c) Constraint:	DPV1 Supported = TRUE
3.4.19.1.	(m) Attribute:	DPV1 Enabled
3.4.19.2.	(m) Attribute:	Fail Safe Required
3.4.19.3.	(m) Attribute:	Fail Safe
3.4.19.4.	(m) Attribute:	WD Base
3.4.20.	(m) Attribute:	List of Real Cfg Data
3.4.20.1.	(s) Attribute:	Cfg Identifier
3.4.20.1.1.	(m) Attribute:	Type
3.4.20.1.2.	(m) Attribute:	Format
3.4.20.1.3.	(m) Attribute:	Consistency
3.4.20.1.4	(m) Attribute:	Length
3.4.20.2.	(s) Attribute:	Special Cfg Identifier
3.4.20.2.1.	(m) Attribute:	Type
3.4.20.2.2.	(c) Constraint:	Type = Input or Type = Input-Output
3.4.20.2.2.1.	(m) Attribute:	Input Length
3.4.20.2.2.2.	(m) Attribute:	Input Format
3.4.20.2.2.3.	(m) Attribute:	Input Consistency
3.4.20.2.3.	(c) Constraint:	Type = Output or Type = Input-Output
3.4.20.2.3.1.	(m) Attribute:	Output Length
3.4.20.2.3.2.	(m) Attribute:	Output Format
3.4.20.2.3.3.	(m) Attribute:	Output Consistency
3.4.20.2.4.	(o) Attribute:	List of Data Description
3.4.20.2.4.1.	(m) Attribute:	Data Description
3.4.20.2.5.	(o) Attribute:	Manufacturer Specific Data
3.4.21.	(o) Attribute:	No Add Change
3.4.22.	(m) Attribute:	Publisher Supported
3.4.23.	(m) Attribute:	Isochronous Mode Supported
3.4.24.	(c) Constraint:	Isochronous Mode Supported = TRUE
3.4.24.1.	(m) Attribute:	Isochronous Mode
3.4.25.	(m) Attribute:	Prm Command Supported
3.5.	(c) Constraint:	AR Type = MS1
3.5.1	(m) Attribute:	Pull Plug Alarm supported
3.5.2	(c) Constraint:	Pull Plug Alarm supported = TRUE
3.5.2.1	(m) Attribute:	Pull Plug Alarm enabled

3.5.3.	(m)	Attribute:	Process Alarm supported
3.5.4.	(c)	Constraint:	Process Alarm supported = TRUE
3.5.4.1.	(m)	Attribute:	Process Alarm enabled
3.5.5.	(m)	Attribute:	Diagnosis Alarm supported
3.5.6.	(c)	Constraint:	Diagnosis Alarm supported = TRUE
3.5.6.1.	(m)	Attribute:	Diagnosis Alarm enabled
3.5.7.	(m)	Attribute:	Manufacturer Specific Alarm supported
3.5.8.	(c)	Constraint:	Manufacturer Specific Alarm supported = TRUE
3.5.8.1.	(m)	Attribute:	Manufacturer Specific Alarm enabled
3.5.9.	(m)	Attribute:	Status Alarm supported
3.5.10.	(c)	Constraint:	Status Alarm supported = TRUE
3.5.10.1.	(m)	Attribute:	Status Alarm enabled
3.5.11.	(m)	Attribute:	Update Alarm supported
3.5.12.	(c)	Constraint:	Update Alarm supported = TRUE
3.5.12.1.	(m)	Attribute:	Update Alarm enabled
3.5.13.	(m)	Attribute:	Alarm Mode supported
3.5.14.	(c)	Constraint:	Alarm Mode supported = TRUE
3.5.14.1.	(m)	Attribute:	Alarm Mode enabled
3.6.	(c)	Constraint:	AR Type = MS3
3.6.1.	(m)	Attribute:	Clock Sync Interval
3.6.2.	(o)	Attribute:	CS Delay Time
3.7.	(o)	Attribute:	AR Name

SERVICES:

1.	(m)	MgtService:	Load ARL
2.	(o)	MgtService:	Get ARL
3.	(m)	MgtService:	Set Cfg
4.	(o)	MgtService:	Set ARL Isochronous Mode

6.2.10.3.1.2 Attributes

Implicit
The attribute Implicit indicates that the ARL DP-slave object is implicitly addressed by the service.

Min Send Timeout
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute defines the minimum Send Timeout common for all MS2 ARs supported in this station.

Attribute Type: Unsigned16

Allowed value: 1 to $2^{16} - 1$

Time Base: 10ms

List of Resource Manager Entries
This conditional attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute is common for all MS2 ARs supported in this station and composed of the following list elements(for each supported MS2 AR a complete list of elements is needed):

API
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. The attribute API specifies the numeric Identifier for the related Application Process of this MS2 AR.

Attribute Type: Unsigned8

Allowed values: 0 to 255 (0 = Default AP)

SCL
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. The attribute SCL specifies the configured Security Level for the identified Application Process of this MS2 AR.

Attribute Type: Unsigned8

Allowed values: 0 to 255 (0 = no Security Level used)

Max Len Data Unit
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute reflects the upper limit of the parameter Length of the services of the Process Data ASE for this MS2 AR.

Attribute Type: Unsigned8

Allowed values: 0 to 240

Send Timeout
This conditional and dynamic attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute defines the Send Timeout for this MS2 AR. This attribute is set to the maximum of the value of the attribute Min Send Timeout and the field Send Timeout of the previously received Initiate-REQ-PDU for this MS2 AR. This attribute is not influenced by the Load ARL service.

Attribute Type: Unsigned16

Allowed value: 1 to 2^{16} - 1

Time Base: 10ms

AREP
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute specifies the local identifier of this MS2 AR.

Attribute Type: Unsigned8

S_SAP_index
This conditional and dynamic attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute contains the assigned S_SAP_index from the list of S_SAP_indices after reception of an Initiate-REQ-PDU for this MS2 AR. This attribute is not influenced by the Load ARL service.

Attribute Type: Unsigned8

Allowed values: 0 to 48

In Use
This conditional and dynamic attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute specifies the state of this MS2 AR. This attribute is set to TRUE after reception of an Initiate-REQ-PDU and assignment of a S_SAP_index for this MS2 AR. This attribute is set to FALSE after release of this MS2 AR. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: FALSE, TRUE

List of S_SAP_indices
This conditional attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute is common for all MS2 ARs supported in this station and composed of the following list elements:

S_SAP_index
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS2. This attribute specifies an available Responder DLSAP for the MS2 ARs of the DP-slave.

Attribute Type: Unsigned8

Allowed values: 0 to 48

List of ARL Entries
This attribute is composed of the following list elements:

AREP
This static attribute contains the local identifier of this AR.

Attribute Type: Unsigned8

AR Type
This static attribute specifies the type of this AR.

Attribute Type: Unsigned8

The allowed values are shown in Table 200.

Table 200 – AR type

Value	Meaning
0	MS0
1	MS1
2	MS2
3	MS3

List of related CRL Entries
This attribute is composed of the following list elements:

CREP
This static attribute contains the local identifier for the related CR for this AR.

Attribute Type: Unsigned8

AR Type = MS0
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MS0.

Prm Req
This dynamic attribute indicates if the DP-slave has to be reparameterized and reconfigured. This attribute shall be set to TRUE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Prm Fault
This dynamic attribute indicates the result of the check of the parameterization. This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Cfg Fault
This dynamic attribute indicates the result of the check of the configuration data. This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Not Supported
This dynamic attribute indicates that a functionality requested by the DP-master (Class 1) is not supported on the MS0 AR by the DP-slave. This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Station Not Ready
This dynamic attribute indicates the readiness to enter the data exchange mode with the assigned DP-master (Class 1). This attribute shall be set to TRUE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Stat Diag
This dynamic attribute indicates that the DP-master (Class 1) shall fetch permanently diagnosis from the DP-slave until the application indicates readiness for operation with the Application Ready service. This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine and the Application Ready service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Master Add
This dynamic attribute contains the DL-address of the DP-master (Class 1) which has locked the DP-slave. This attribute shall be set to invalid (= 255) with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Unsigned8

Allowed values: 0 to 126, 255

Sync Supported
This static attribute reflects the ability of a DP-slave to participate in the synchronization of the Output Data of all or a group of DP-slaves. As the check whether the Sync ability is supported is done during parameterization phase, errors will be avoided during the data exchange mode. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Table 201 shows the allowed values for this attribute.

Table 201 – Sync supported

Value	Meaning
FALSE	The DP-slave does not support the Sync ability
TRUE	The DP-slave supports the Sync ability

Sync Mode
This dynamic attribute indicates if the Sync mode is activated. This attribute shall always be set to FALSE if the attribute Sync Supported has the value FALSE. If the attribute Sync Supported has the value TRUE the following rules shall apply for the attribute Sync Mode:

- It shall be set to TRUE as soon as the control command Sync is received, which is conveyed by the Global Control service.
- It shall be set to FALSE as soon as the control command Unsync is received which is conveyed by the Global Control service.

This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Freeze Supported
This static attribute reflects the ability of a DP-slave to participate in the synchronization of the Input Data of all or a group of DP-slaves. As the check whether the Freeze ability is supported is done during parameterization phase, errors will be avoided during the data exchange mode. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Table 202 shows the allowed values for this attribute.

Table 202 – Freeze supported

Value	Meaning
FALSE	The DP-slave does not support the Freeze ability
TRUE	The DP-slave supports the Freeze ability

Freeze Mode
This dynamic attribute indicates if the Freeze mode is activated. This attribute shall always be set to FALSE if the attribute Freeze Supported has the value FALSE. If the attribute Freeze Supported has the value TRUE the following rules shall apply for the attribute Freeze Mode:

- It shall be set to TRUE as soon as the control command Freeze is received, which is conveyed by the Global Control service.
- It shall be set to FALSE as soon as the control command Unfreeze is received which is conveyed by the Global Control service.

This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine

Attribute Type: Boolean.

WD Enabled
This dynamic attribute indicates if the watchdog control is activated. This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

WD Fact 1, WD Fact 2
These dynamic attributes contain the values for the monitoring of the DP-master (Class 1) at the MS0 AR (watchdog control (T_{WD})). The watchdog control in a DP-slave takes care that, if the DP-master fails, the outputs will be set in the safe state after the expiration of this time. The time is calculated according to equation:

T_{WD} = Watchdog Time Base × WD Fact 1 × WD Fact 2

If the attribute DPV1 Supported has the value FALSE the Watchdog Time Base shall be set to 10 ms.

If the attribute DPV1 Supported has the value TRUE the Watchdog Time Base shall be set to the value 1ms if the attribute WD Base has the value TRUE. If the attribute DPV1 Supported has the value TRUE the Watchdog Time Base shall be set to the value 10ms if the attribute WD Base has the value FALSE. These attributes are influenced by the MS0 AR state machine. These attributes are not influenced by the Load ARL service.

Attribute Type: Unsigned8

Allowed values: 1 to 255

Min TSDR
This dynamic attribute contains the value for the Minimum Station Delay Responder time of the DLL. If this Parameter is equal zero, the previous value of the Min TSDR remains unchanged. This attribute is influenced by the MS0 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Unsigned8

Allowed values: 0 to 255

Ident Number
This static attribute contains the Ident Number of the DP-slave. The DP-slave accepts the parameterization and change of the DL-address only if the transmitted Ident Number in the correspondent service is identical with this attribute. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Unsigned16

Allowed values: 0 to 2^{16} - 1

Group Identifier
This dynamic attribute indicates the group(s) to which this DP-slave has been assigned by the DP-master (Class 1) in the parameterization. The Group Identifier distinguishes 8 groups. If the Group Identifier is set to No Group, the DP-slave is not assigned to any group. This attribute is influenced by the MS0 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: OctetString(1)

The allowed values are shown in Table 203.

Table 203 – Group identifier

Value	Meaning
Bit 0 = 0	Group 1 has no access
Bit 0 = 1	Group 1 has access
Bit 1 = 0	Group 2 has no access
Bit 1 = 1	Group 2 has access
Bit 2 = 0	Group 3 has no access
Bit 2 = 1	Group 3 has access
Bit 3 = 0	Group 4 has no access
Bit 3 = 1	Group 4 has access
Bit 4 = 0	Group 5 has no access
Bit 4 = 1	Group 5 has access
Bit 5 = 0	Group 6 has no access
Bit 5 = 1	Group 6 has access
Bit 6 = 0	Group 7 has no access
Bit 6 = 1	Group 7 has access
Bit 7 = 0	Group 8 has no access
Bit 7 = 1	Group 8 has access

DPV1 Supported
This static attribute indicates whether the DP-slave supports the extended functionality. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

DPV1 Enabled
This dynamic attribute indicates if the extended functionality is activated. This conditional attribute shall only be present if the attribute DPV1 Supported has the value TRUE. The following rules shall apply for the attribute DPV1 Enabled:

- It shall be set to TRUE if the DP-master (Class 1) requests the DPV1 support in the parameterization.
- It shall be set to FALSE if the DP-master (Class 1) requests no DPV1 support in the parameterisation.

This attribute shall be set to FALSE with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Table 204 shows the allowed values for this attribute.

Table 204 – DPV1 enabled

Value	Meaning
FALSE	The extended functionality related with the attribute DPV1 Supported is disabled.
TRUE	The extended functionality related with the attribute DPV1 Supported is enabled.

Fail Safe Required
This static attribute indicates if the DP-slave requires the Fail Safe mode for operation. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Fail Safe
This dynamic attribute indicates if the DP-master (Class 1) supports the Fail Safe mode. If the DP-slave requires the Fail Safe mode and the DP-master (Class 1) does not support this mode the MS0 AR state machine shall set the attribute Prm Fault to TRUE in the MS0 AR entry. This attribute is influenced by the MS0 AR state machine. This attribute is not influenced by the Load ARL service.

Table 205 shows the allowed values for this attribute.

Table 205 – Fail safe

Value	Meaning
FALSE	The DP-master (Class 1) does not support the Fail Safe mode.
TRUE	The DP-master (Class 1) supports the Fail Safe mode.

WD Base
This dynamic attribute contains the Time Base value for the watchdog control according to the parameterization. This attribute is influenced by the MS0 AR state machine. This attribute is not influenced by the Load ARL service.

Table 206 shows the allowed values for this attribute.

Table 206 – WD base

Value	Meaning
FALSE	The Watchdog Time Base is 10ms.
TRUE	The Watchdog Time Base is 1ms.

List of Real Cfg Data
This dynamic attribute is composed of the following list elements:

Cfg Identifier
This dynamic attribute describes the basic configuration format and shall be set with the Load ARL service or with the Set Cfg service. This attribute is composed of the following elements:

Type
This dynamic attribute indicates the type of the related Input/Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Input, Output, Input-Output

Format
This dynamic attribute indicates the format of the related Input/Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Byte, Word

Consistency
This dynamic attribute indicates the consistency of the related Input/Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Byte, Word, whole length

Length
This dynamic attribute indicates the length of the related Input/Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: 1 to 32

Special Cfg Identifier
This dynamic attribute describes the special configuration format and is composed of the following elements. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Type
This dynamic attribute indicates the type of the related Input/Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Input, Output, Input-Output

Input Length
This dynamic attribute indicates the length of the related Input Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Attribute Type: Unsigned8

Allowed values: 1 to 128

Input Format
This dynamic attribute indicates the format of the related Input Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Byte, Word

Input Consistency
This dynamic attribute indicates the consistency of the related Input Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Byte, Word, whole length

Output Length
This dynamic attribute indicates the length of the related Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Attribute Type: Unsigned8

Allowed values: 1 to 128

Output Format
This dynamic attribute indicates the format of the related Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Byte, Word

Output Consistency
This dynamic attribute indicates the consistency of the related Output Data Element. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Allowed values: Byte, Word, whole length

List of Data Description
This dynamic attribute is composed of the following list elements. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Data Description
This dynamic attribute contains one of the allowed Data Types of the data description of one element of the Extended Input/Output Data object. The allowed Data Types are given in IEC 61158-5-10, Clause 5. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

Manufacturer Specific Data
This dynamic attribute contains the manufacturer specific data for one Special Cfg Identifier. This attribute shall be set accordingly with the Load ARL service. Furthermore the attribute is influenced by the Set Cfg service.

No Add Change
This dynamic attribute indicates whether it is allowed to change the DP-slave address. To change this attribute itself an initial reset of the DP-slave is necessary. After the initial reset the DP-slave takes the default address 126 and the attribute No Add Change will be set to FALSE. This attribute shall be set accordingly with the Load ARL service. This attribute is further influenced by the MS0 AR state machine.

Table 207 shows the allowed values for this attribute.

Table 207 – No Add change

Value	Meaning
FALSE	Change of the address is allowed
TRUE	Change of the address is not allowed

Publisher Supported
This static attribute indicates the capability of a DP-slave to operate as a Publisher. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Isochronous Mode Supported
This static attribute indicates the capability of a DP-slave to operate in the Isochronous Mode. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Isochronous Mode
This dynamic attribute indicates if the DP-slave shall operate in the Isochronous Mode. This attribute shall be set to FALSE with the Load ARL service. After reception of the IsoM Parameter with the Check User Prm or Check Ext User Prm service the User shall set this attribute accordingly with the Set ARL Isochronous Mode service. This shall be done before requesting the services Check User Prm Result or Check Ext User Prm Result.

This conditional attribute shall only be present if the attribute Isochronous Mode Supported is set to TRUE. This attribute is not influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

AR Type = MS1
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MS1. This attribute shall furthermore only be present if the attribute DPV1 Supported is set to TRUE in the MS0 AR entry.

Pull Plug Alarm supported
This static attribute indicates the capability of a DP-slave to send a Pull or a Plug Alarm. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Pull Plug Alarm enabled
This dynamic attribute indicates if the transmission of the Pull Plug Alarm is enabled by the DP-master (Class 1). This conditional attribute shall only be present if the attribute Pull Plug Alarm supported is set to TRUE. This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Process Alarm supported
This static attribute indicates the capability of a DP-slave to send a Process Alarm. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Process Alarm enabled
This dynamic attribute indicates if the transmission of the Process Alarm is enabled by the DP-master (Class 1). This conditional attribute shall only be present if the attribute Process Alarm supported is set to TRUE. This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Diagnosis Alarm supported
This static attribute indicates the capability of a DP-slave to send a Diagnosis Alarm. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Diagnosis Alarm enabled
This dynamic attribute indicates if the transmission of the Diagnosis Alarm is enabled by the DP-master (Class 1). This conditional attribute shall only be present if the attribute Diagnosis Alarm supported is set to TRUE. This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Manufacturer Specific Alarm supported
This static attribute indicates the capability of a DP-slave to send a Manufacturer Specific Alarm. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Manufacturer Specific Alarm enabled
This dynamic attribute indicates if the transmission of the Manufacturer Specific Alarm is enabled by the DP-master (Class 1). This conditional attribute shall only be present if the attribute Manufacturer Specific Alarm supported is set to TRUE. This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Status Alarm supported
This static attribute indicates the capability of a DP-slave to send a Status Alarm. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Status Alarm enabled
This dynamic attribute indicates if the transmission of the Status Alarm is enabled by the DP-master (Class 1). This conditional attribute shall only be present if the attribute Status Alarm supported is set to TRUE. This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Update Alarm supported
This static attribute indicates the capability of a DP-slave to send a Update Alarm. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Update Alarm enabled
This dynamic attribute indicates if the transmission of the Update Alarm is enabled by the DP-master (Class 1). This conditional attribute shall only be present if the attribute Update Alarm supported is set to TRUE. This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Alarm Mode supported
This static attribute indicates the maximum number of possible active alarms in a DP-slave. This attribute shall be set accordingly with the Load ARL service.

Table 208 shows the allowed values for this attribute.

Table 208 – Alarm mode supported

Value	Meaning
0	1 alarm of each type
1	2 alarms in total
2	4 alarms in total
3	8 alarms in total
4	12 alarms in total
5	16 alarms in total
6	24 alarms in total
7	32 alarms in total

Alarm Mode enabled
This dynamic attribute indicates the maximum number of possible active alarms negotiated with the DP-master (Class 1) on this MS1 AR. The negotiation shall be done according to the following rules:

- If the attribute Alarm Mode is set to the value 0 and the desired Alarm Mode of the DP-master (Class 1) contained in the parameterization is set to 0 than this attribute shall be set to 0.
- If the attribute Alarm Mode is set to the value 0 and the desired Alarm Mode of the DP-master (Class 1) contained in the parameterization is set to a value different from 0 than this attribute shall not be set.
- If the attribute Alarm Mode is set to a value different from 0 and the desired Alarm Mode of the DP-master (Class 1) contained in the parameterization is set to 0 than this attribute shall be set to 0.

If the attribute Alarm Mode is set to a value different from 0 and the desired Alarm Mode of the DP-master (Class 1) contained in the parameterisation is set to a value different from 0 than this attribute shall be set to the lesser value of both.

This attribute is influenced by the MS1 AR state machine. This attribute is not influenced by the Load ARL service.

Attribute Type: Unsigned8

Allowed values: see Table 208

AR Type = MS3
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MS3. This attribute shall furthermore only be present if the attribute DPV1 Supported is set to TRUE in the MS0 AR entry.

Clock Sync Interval
This static attribute contains the value for the clock synchronization interval.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 10ms

The default value shall be 10-s.

CS Delay Time
This static and optional attribute contains the transmission delay of the network (line delay, repeater delay, ...). It should be included for the calculation of the parameter Time Value of the Set Time service.

Attribute Type: Network Time Difference

The default value shall be 0.

AR Name
This static attribute contains the name of the related ARL Entry. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Visible String (32)

6.2.10.3.1.3 Invocation of ARL DP-slave object

For the invocation of the ARL DP-slave object the following rules apply:

- Only one ARL DP-slave object shall be invoked in a DP-slave.
- Only one MS0 Entry shall be contained in the List of ARL Entries
- Only one MS1 Entry shall be contained in the List of ARL Entries
- Only one MS3 Entry shall be contained in the List of ARL Entries

No MM1 or MM2 Entry shall be contained in the List of ARL Entries

6.2.10.3.2 ARL DP-master (class 1) class specification

6.2.10.3.2.1 Template

The ARL DP-master (Class 1) object is described by the following template:

DP ASE:			**AR ASE**
CLASS:		**ARL DP-master (Class 1)**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(c)	Constraint:	at least one entry with AR Type = MS1
2.1.	(m)	Attribute:	Alarm Max
3.	(m)	Attribute:	List of ARL Entries
3.1.	(m)	Attribute:	AREP
3.2.	(m)	Attribute:	AR Type
3.3.	(m)	Attribute:	List of related CRL Entries
3.3.1.	(m)	Attribute:	CREP
3.4.	(c)	Constraint:	AR Type = MS0
3.4.1.	(m)	Attribute:	Error Action Flag
3.4.2.	(m)	Attribute:	Min Slave Interval
3.4.3.	(m)	Attribute:	Data Control Time
3.4.4.	(m)	Attribute:	Max User Global Control
3.4.5.	(m)	Attribute:	Max Diag Data Len
3.4.5.	(m)	Attribute:	Isochronous Mode supp
3.4.6.	(c)	Constraint:	Isochronous Mode supp = TRUE
3.4.6.1.	(m)	Attribute:	Isochronous Mode
3.4.6.2.	(m)	Attribute:	IsoM Freeze
3.4.6.3.	(m)	Attribute:	IsoM Sync
3.4.6.4.	(m)	Attribute:	T_{CT}
3.4.6.5.	(m)	Attribute:	$maxT_{SH}$
3.5.	(c)	Constraint:	AR Type = MS1
3.5.2.	(m)	Attribute:	Extra Alarm SAP
3.5.3.	(m)	Attribute:	Max Channel Data Length
3.5.4.	(m)	Attribute:	Alarm Mode
3.5.5.	(m)	Attribute:	MS1 Timeout
3.6.	(c)	Constraint:	AR Type = MM1
3.6.1.	(m)	Attribute:	Poll Timeout
3.6.2.	(m)	Attribute:	Ident Number
3.7.	(c)	Constraint:	AR Type = MS3
3.7.1.	(m)	Attribute:	Time Device Type

3.7.2.	(c)	Constraint:	Time Device Type = Time Receiver
3.7.2.1.	(m)	Attribute:	Clock Sync Interval
3.7.2.2.	(o)	Attribute:	CS Delay Time
3.8.	(o)	Attribute:	AR Name

SERVICES:

1.	(m)	MgtService:	Load ARL
2.	(o)	MgtService:	Get ARL
3.	(o)	MgtService:	ARL Slave Update DP-master CI1

6.2.10.3.2.2 Attributes

Implicit
The attribute Implicit indicates that the ARL DP-slave object is implicitly addressed by the service.

Alarm Max
This conditional and static attribute shall only be present if the List of ARL Entries contains at least one entry with the attribute AR Type set to MS1. It indicates the maximum number of alarms per DP-slave, which can be handled by the DP-master (Class 1). It describes a Master capability and is common for all MS1 ARs. The Master shall be able to handle a minimum number of 7 alarms, because there are existing 7 different alarm types. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned8

Allowed values: 7 to 32

List of ARL Entries
This attribute is composed of the following list elements:

AREP
This static attribute contains the local identifier of this AR.

AR Type
This static attribute specifies the type of this AR.

Allowed values: MS0, MS1, MS3, MM1, MM2

List of related CRL Entries
This attribute is composed of the following list elements:

CREP
This static attribute contains the local identifier for the related CR for this AR.

AR Type = MS0
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MS0.

Error Action Flag
This static attribute indicates how the reaction of the DP-master (Class 1) in case of failed data transfers in the operation mode OPERATE shall be. The reaction shall be performed according to the following rules:

- If the data transfer to at least one selected (dependent on the attribute Ignore AClr in the List of CRL Entries) DP-slave was erroneous and the Error Action Flag is set to TRUE, the operation mode will change from OPERATE to CLEAR.

If the Error Action Flag is set to FALSE, the DP-master (Class 1) will remain in the operation mode OPERATE despite erroneous data transfers.

Furthermore this attribute indicates how the transition from the operation mode CLEAR to the operation mode OPERATE shall be performed:

- If the Error Action Flag set to TRUE the transition shall only be performed if the data transfer to the selected DP-slaves has been successfully executed.

If the Error Action Flag set to FALSE the transition shall be performed independently of a successfully executed data transfer to all activated DP-slaves.

This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Boolean

Allowed values: TRUE, FALSE

Min Slave Interval
This static attribute indicates the smallest allowed period of time between two consecutive DP-slave poll cycles. This ensures that the sequence of cyclic service requests from the DP-master (Class 1) can be handled by the DP-slave. This period of time will be complied by the DP-master (Class 1) at the beginning of each cyclic data transfer. This time interval is not guaranteed for the User Global Control services and for acyclic MS1 AR requests. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned16

Allowed values: 1 to $2^{16} - 1$

Time Base: 100µs

Data Control Time
This static attribute indicates the period of time in which a Global Control shall be sent indicating the operation mode of the DP-master (Class 1) and at least one successful interaction shall be executed with every activated DP-slave. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned16

Allowed values: 1 to $2^{16} - 1$

Time Base: 10ms

Max User Global Control
This static attribute indicates the maximum number of Global Control requests, which may be submitted by the User before the Global Control confirmations have been received. This attribute describes the ability of the DP-master (Class 1). A practicable value for Max User Global Control is 16. For each of the 8 Slave groups one Sync and one Freeze Command may be started at the same time. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned8

Allowed values: 1 to 255

Max Diag Data Len
This static attribute indicates the maximum length of Diag Data of all assigned DP-slaves. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned8

Allowed values: 6 to 244

Isochronous Mode supp
This static attribute indicates the capability of the DP-master (Class 1) concerning the Isochronous Mode. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Boolean

The allowed values are shown Table 209.

Table 209 – Isochronous mode supp

Value	Meaning
FALSE	Isochronous Mode not supported by the DP-master (Class 1)
TRUE	Isochronous Mode supported by the DP-master (Class 1)

Isochronous Mode
This attribute indicates if the DP-master (Class 1) shall operate in the Isochronous Mode. Two different models concerning the interface between the DP Application Layer and the application of the DP-master (Class 1) itself are provided by fieldbus DP for the DP-master (Class 1), see 6.2.2.4.3. This attribute shall be set accordingly with the Load ARL service.

The allowed values are shown in Table 210.

Table 210 – Isochronous mode

Value	Meaning
Not Synchronized	the DP-master (Class 1) shall not operate in the Isochronous Mode
Buffered Synchronized	the DP-master (Class 1) shall operate in the Buffer Synchronized Isochronous Mode
Enhanced Synchronized	the DP-master (Class 1) shall operate in the Enhanced Synchronized Isochronous Mode

IsoM Freeze
This attribute indicates if the synchronization message (SYNCH) shall be sent with the Control Command Freeze set to TRUE. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Boolean

Allowed values: TRUE, FALSE

IsoM Sync
This attribute indicates if the synchronization message (SYNCH) shall be sent with the Control Command Sync set to TRUE. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Boolean

Allowed values: TRUE, FALSE

T_{CT}
This attribute specifies the time (in units of t_{BIT}) that is necessary to handle a complete Isochronous DP cycle and corresponds to the time T_{DP}. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned32

Allowed values: 1 to $2^{24} - 1$

maxT_{SH}
This attribute specifies the maximum allowed shift (in units of t_{BIT}) of the time T_{DP} measured by the DP-master (Class 1). The application of the DP-master (Class 1) will be informed if the measured Isochronous DP cycle time exceeds this limit by the service SYNCH Delayed indicating the difference. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned8

Allowed values: 1 to 2^8 - 1

AR Type = MS1
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MS1.

Extra Alarm SAP
This static attribute selects the CR for the Alarm Ack service. If the attribute is set to FALSE the DP-master (Class 1) acknowledges alarms via the CR for the read write services. If the attribute is set to TRUE the DP-master (Class 1) acknowledges alarms via a different CR. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Boolean

Allowed values: TRUE, FALSE

Max Channel Data Length
This static attribute indicates the maximum length of the MS1 ARDUs (Length parameter and 4 Byte Header) of the corresponding DP-slave. This attribute shall be set accordingly with the Load ARL service.

Attribute Type : Unsigned8

Allowed values: 4 to 244

Alarm Mode
This static attribute indicates the maximum number of possible active alarms of the corresponding DP-slave. This attribute shall be set accordingly with the Load ARL service.

Table 211 shows the allowed values for this attribute.

Table 211 – Alarm mode

Value	Meaning
0	1 alarm of each type
1	2 alarms in total
2	4 alarms in total
3	8 alarms in total
4	12 alarms in total
5	16 alarms in total
6	24 alarms in total
7	32 alarms in total

MS1 Timeout
This attribute contains the value for the Timeout of the acyclic communication on the MS1 AR and specifies the maximum period of time it may take a DP-slave to provide a response. The time contained in this attribute is the maximum time of the assigned DP-slaves.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 10ms

AR Type = MM1
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MM1.

Poll Timeout
This static attribute indicates the maximum period of time in which the requester shall fetch the response. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 1ms

Ident Number
This static attribute contains the Ident Number of the DP-master (Class 1). This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

AR Type = MS3
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MS3.

Time Device Type
This static attribute indicates the Device Type concerning the Time ASE. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Unsigned8

The allowed values are shown in Table 212.

Table 212 – Time device type

Value	Meaning
0	Time Receiver
1	Time Master

Clock Sync Interval
This static and conditional attribute contains the value for the clock synchronization interval. It shall only be present if the attribute Time Device Type has the value Time Receiver.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 10ms

The default value shall be 10 s.

CS Delay Time
This static, conditional and optional attribute contains the transmission delay of the network (line delay, repeater delay, …). It should be included for the calculation of the parameter Time Value of the Set Time service. It shall only be present if the attribute Time Device Type has the value Time Receiver.

Attribute Type: Network Time Difference

The default value shall be 0.

AR Name
This static attribute contains the name of the related ARL Entry. This attribute shall be set accordingly with the Load ARL service.

6.2.10.3.2.3 Invocation of ARL DP-master (class 1) object

For the invocation of the ARL DP-master (Class 1) object the following rules apply:

- Only one ARL DP-master (Class 1) object shall be invoked in a DP-master (Class 1).
- Only one MS0 Entry shall be contained in the List of ARL Entries.
- Only one MS3 Entry shall be contained in the List of ARL Entries.
- Only one MM1 Entry shall be contained in the List of ARL Entries.
- Only one MM2 Entry shall be contained in the List of ARL Entries.

No MS2 Entry shall be contained in the List of ARL Entries.

6.2.10.3.3 ARL DP-master (class 2) class specification

6.2.10.3.3.1 Template

The ARL DP-master (Class 2) object is described by the following template:

DP ASE:			**AR ASE**
CLASS:		**ARL DP-master (Class 2)**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of ARL Entries
2.1.	(m)	Attribute:	AREP
2.2.	(m)	Attribute:	AR Type
2.3.	(m)	Attribute:	List of related CRL Entries
2.3.1.	(m)	Attribute:	CREP
2.4.	(c)	Constraint:	AR Type = MM1
2.4.1	(m)	Attribute:	Poll Timeout
2.5	(o)	Attribute:	AR Name
SERVICES:			
1.	(m)	MgtService:	Load ARL
2.	(o)	MgtService:	Get ARL

6.2.10.3.3.2 Attributes

Implicit
The attribute Implicit indicates that the ARL DP-master (Class 2) object is implicitly addressed by the service.

List of ARL Entries
This attribute is composed of the following list elements:

AREP
This static attribute contains the local identifier of this AR.

AR Type
This static attribute specifies the type of this AR.

Allowed values: MS0, MS2, MM1, MM2

List of related CRL Entries
This attribute is composed of the following list elements:

CREP
This static attribute contains the local identifier for the related CR for this AR.

AR Type = MM1
This constraint reflects the attributes for this AR if the value of the attribute AR Type is set to MM1.

Poll Timeout
This static attribute indicates the maximum period of time in which the responder shall provide the response. This attribute shall be set accordingly with the Load ARL service.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

Time Base: 1ms

AR Name
This static attribute contains the name of the related ARL Entry. This attribute shall be set accordingly with the Load ARL service.

6.2.10.3.3.3 Invocation of ARL DP-master (class 2) object

For the invocation of the ARL DP-master (Class 2) object the following rules apply:

- Only one ARL DP-master (Class 2) object shall be invoked in a DP-master (Class 2).
- Only one MM1 Entry shall be contained in the List of ARL Entries.
- Only one MM2 Entry shall be contained in the List of ARL Entries.

No MS1, MS3 Entry shall be contained in the List of ARL Entries.

6.2.10.4 Communication relationship list class specification

6.2.10.4.1 CRL DP-slave class specification

6.2.10.4.1.1 Template

The CRL DP-slave object is described by the following template:

DP ASE:			**AR ASE**
CLASS:		**CRL DP-slave**	
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of CRL Entries
2.1.	(m)	Attribute:	CREP
2.2.	(m)	Attribute:	CR Type
2.3.	(m)	Attribute:	Related AREP
2.4.	(m)	Attribute:	S_SAP_index
2.5.	(m)	Attribute:	D_addr
2.6.	(m)	Attribute:	Service_activate
2.7.	(m)	Attribute:	Role_in_Service
2.8.	(c)	Constraint:	Service_activate = SRD or MSRD
2.8.1	(m)	Attribute:	Indication_mode
2.9.	(m)	Attribute:	Max_DLSDU_length_req_low
2.10.	(m)	Attribute:	Max_DLSDU_length_req_high
2.11.	(m)	Attribute:	Max_DLSDU_length_ind_low
2.12.	(m)	Attribute:	Max_DLSDU_length_ind_high
2.13.	(c)	Constraint:	CR Type = BBUU-OM & Role_in_Service = Initiator or Both
2.13.1.	(m)	Attribute:	Publisher Address
2.13.2.	(m)	Attribute:	Input Data Length Publisher
2.13.3.	(m)	Attribute:	Source Offset Publisher
2.13.4.	(m)	Attribute:	Length
2.14.	(o)	Attribute:	CR Name
SERVICES:			
1.	(m)	MgtService:	Load CRL
2.	(o)	MgtService:	Get CRL
3.	(o)	MgtService:	Load CRL DXB Link Entries

6.2.10.4.1.2 Attributes

Implicit
The attribute Implicit indicates that the CRL DP-slave object is implicitly addressed by the service.

List of CRL Entries
This attribute is composed of the following list elements:

CREP
This static attribute contains the local identifier of this CR. This attribute shall be set accordingly with the Load CRL service.

CR Type
This static attribute indicates the type of this CR. This attribute is dependent of the AR Type of the related AR. This attribute shall be set accordingly with the Load CRL service. For each AR Type the correspondent CRL entries shall be set.

Table 213 shows the allowed values for this attribute.

Related AREP
This static attribute contains the local identifier for the related AR for this CR. This attribute shall be set accordingly with the Load CRL service.

S_SAP_index
This static attribute contains the identifier for the related Service Access Point of the local data-link layer. The value for this attribute has to be set correspondent to the attributes CR Type of this CR and to the attribute AR Type of the related AR. This attribute shall be set accordingly with the Load CRL service. Table 213 shows the allowed values for this attribute.

Table 213 – S_SAP_index

AR Type	CR Type	S_SAP_index	Remark
MS0	BBUB	NIL	Mandatory for the MS0 AR
MS0	BBUU-OM	NIL	Optional for the MS0 AR
MS0	BQUU	61, 62	Mandatory for the MS0 AR
MS0	BQUU	53	Optional for the MS0 AR
MS0	BBUU	60	Mandatory for the MS0 AR
MS0	QQUU-OM	58	Mandatory for the MS0 AR
MS0	QBCB	55, 56, 57, 59, 60	Mandatory for the MS0 AR
MS1	QQCB-CO	50	Optional for the MS1 AR
MS1	QQCB-CO	51	Mandatory for the MS1 AR
MS1	QQUU-CO	60	Mandatory for the MS1 AR
MS2	QQCB-CO	0 to 48	Mandatory for each MS2 AR
MS2	QQUU-CO	49	Mandatory for the MS2 AR
MS3	QQUU-OM	CS	Mandatory for the MS3 AR

D_addr
This dynamic attribute contains the DL-address of the remote DP-master or Publisher. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index of this CR and to the attribute AR Type of the related AR. Table 214 shows the allowed values for this attribute used with the Load CRL service. Furthermore this attribute is influenced by the AR state machine dependent of the attribute CR Type of this CR and of the attribute AR Type of the related AR. Table 214 shows the allowed values for this attribute for the modification of the AR state machine.

Table 214 – D_addr

AR Type	CR Type	S_SAP_index	D_addr as loaded with the Load CRL service	D_addr modified by the AR state machine
MS0	BBUB	NIL	None	0 to 125
MS0	BBUU-OM	NIL	None	All
MS0	BQUU	61, 62, 53	All	-
MS0	BBUU	60	All	-
MS0	QQUU-OM	58	None	0 to 125
MS0	QBCB	55, 59, 60	All	-
MS0	QBCB	56, 57	None	All
MS1	QQCB-CO	50, 51	None	0 to 125
MS1	QQUU-CO	60	All	-
MS2	QQCB-CO	0 to 48	None	0 to 125
MS2	QQUU-CO	49	All	-
MS3	QQUU-OM	CS	All	-

Where

None	means that the local Service Access Point shall not be activated
All	means that the local Service Access Point shall be activated with no access protection
0 to 125	means that the local Service Access Point shall be activated with the corresponding access protection
-	means that the AR state machines shall not modify the attribute D_addr

Service_activate

This static attribute indicates the DL services which shall be activated for the S_SAP_index of this CRL entry.

The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 215 shows the allowed values for this attribute.

Table 215 – Service_activate

AR Type	CR Type	S_SAP_index	Service_activate
MS0	BBUB	NIL	(M)SRD
MS0	BBUU-OM	NIL	MSRD
MS0	BQUU	61, 62, 53	SRD
MS0	BBUU	60	SRD
MS0	QQUU-OM	58	SDN
MS0	QBCB	55, 56, 57, 59, 60	SRD
MS1	QQCB-CO	50	SRD
MS1	QQCB-CO	51	SRD
MS1	QQUU-CO	60	SRD
MS2	QQCB-CO	0 to 48	SRD
MS2	QQUU-CO	49	SRD
MS3	QQUU-OM	CS	CS Time Event CS Clock Value

Role_in_service
This dynamic attribute indicates the role which shall be activated for the S_SAP_index of this CRL entry. The following values are specified:

- Initiator: The station initiates the respective service exclusively.
- Responder: The station response to the service exclusively.
- Subscriber: The station receives the reply DLPDU of this service exclusively.
- Both: The station initiates the service and response to the service.

Responder/Subscriber: The station responds to the service and receives the reply DLPDU of this service

The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR.

Table 216 shows the allowed values for this attribute.

Table 216 – Role_in_service

AR Type	CR Type	S_SAP_index	Role_in_service as loaded with the Load CRL service	Role_in_service modified by the AR state machine
MS0	BBUB	NIL	None	Responder
MS0	BBUU-OM	NIL	None	Responder, Subscriber, Responder/Subscriber
MS0	BQUU	61, 62, 53	Responder	-
MS0	BBUU	60	Responder	-
MS0	QQUU-OM	58	None	Responder
MS0	QBCB	55, 59, 60	Responder	-
MS0	QBCB	56, 57	None	Responder
MS1	QQCB-CO	50, 51	None	Responder
MS1	QQUU-CO	60	Responder	-
MS2	QQCB-CO	0 to 48	None	Responder
MS2	QQUU-CO	49	Responder	-
MS3	QQUU-OM	CS	Responder	-

Where

None means that the local Service Access Point shall not be activated

– means that the AR state machines shall not modify the attribute Role_in_service

Indication_mode
This static attribute shall only be present if the attribute Service_activate of this CRL entry has the value SRD or MSRD. It indicates the Indication_mode which the DLL shall use to convey the indications of the (M)SRD service. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 217 shows the allowed values for this attribute.

Table 217 – Indication_mode

AR Type	CR Type	S_SAP_index	Role_in_service	Indication_mode
MS0	BBUB	NIL	Responder	All
MS0	BBUU-OM	NIL	Subscriber	-
MS0	BBUU-OM	NIL	Responder, Responder/Subscriber	All
MS0	BQUU	61, 62, 53	Responder	All
MS0	BBUU	60	Responder	All
MS0	QBCB	55, 56, 57, 59	Responder	Data
MS0	QBCB	60	Responder	All
MS1	QQCB-CO	50, 51	Responder	Data
MS1	QQUU-CO	60	Responder	All
MS2	QQCB-CO	0 to 48	Responder	Data
MS2	QQUU-CO	49	Respondr	Data
where - means that the attribute Indication_mode is not present for this Role_in_service				

Max_DLSDU_length_req_low
This static attribute indicates the maximum length for the DLSDU with low priority as requester. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 218 shows the allowed values for this attribute.

Table 218 – Max_DLSDU_length_req_low

AR Type	CR Type	S_SAP_index	Role_in_service	Max_DLSDU_length_req_low
MS0	BBUB	NIL	Responder	0 to 244
MS0	BBUU-OM	NIL	Subscriber	0
MS0	BBUU-OM	NIL	Responder, Responder/Subscriber	0 to 244
MS0	BQUU	61, 62, 53	Responder	0
MS0	BBUU	60	Responder	6 to 244
MS0	QBCB	55	Responder	0
MS0	QBCB	56, 57, 59	Responder	0 to 244
MS0	QQUU-OM	58	Responder	0
MS1	QQCB-CO	50	Responder	4
MS1	QQCB-CO	51	Responder	4 to 244
MS1	QQUU-CO	60	Responder	6 to 244
MS2	QQCB-CO	0 to 48	Responder	48 to 244
MS2	QQUU-CO	49	Responder	4
MS3	QQUU-OM	CS	Responder	0

Max_DLSDU_length_req_high
This static attribute indicates the maximum length for the DLSDU with high priority as requester. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 219 shows the allowed values for this attribute.

Table 219 – Max_DLSDU_length_req_high

AR Type	CR Type	S_SAP_index	Role_in_service	Max_DLSDU_length_req_high
MS0	BBUB	NIL	Responder	0 to 244
MS0	BBUU-OM	NIL	Subscriber	0
MS0	BBUU-OM	NIL	Responder, Responder/Subscriber	0 to 244
MS0	BQUU	61, 62, 53	Responder	0
MS0	BBUU	60	Responder	0
MS0	QBCB	55, 56, 57, 59	Responder	0
MS0	QQUU-OM	58	Responder	0
MS1	QQCB-CO	50, 51	Responder	0
MS1	QQUU-CO	60	Responder	0
MS2	QQCB-CO	0 to 48	Responder	0
MS2	QQUU-CO	49	Responder	0
MS3	QQUU-CO	CS	Responder	0

Max_DLSDU_length_ind_low
This static attribute indicates the maximum length for the DLSDU with low priority for indication. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index and Role_in_service of this CR and to the attribute AR Type of the related AR.

Table 220 shows the allowed values for this attribute.

Table 220 – Max_DLSDU_length_ind_low

AR Type	CR Type	S_SAP_index	Role_in_service	Max_DLSDU_length_ind_low
MS0	BBUB	NIL	Responder	0
MS0	BBUU-OM	NIL	Subscriber, Responder/Subscriber	244
MS0	BBUU-OM	NIL	Responder	0
MS0	BQUU	61, 62, 53	Responder	0
MS0	BBUU	60	Responder	0
MS0	QBCB	55	Responder	0
MS0	QBCB	56, 57, 59	Responder	0
MS0	QQUU-OM	58	Responder	0
MS1	QQCB-CO	50	Responder	4
MS1	QQCB-CO	51	Responder	4 to 244
MS1	QQUU-CO	60	Responder	0
MS2	QQCB-CO	0 to 48	Responder	4 to 244
MS2	QQUU-CO	49	Responder	48 to 64
MS3	QQUU-OM	CS	Responder	18

Max_DLSDU_length_ind_high
This static attribute indicates the maximum length for the DLSDU with high priority for indication. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and S_SAP_index and Role_in_service of this CR and to the attribute AR Type of the related AR.

Table 221 shows the allowed values for this attribute.

Table 221 – Max_DLSDU_length_ind_high

AR Type	CR Type	S_SAP_index	Role_in_service	Max_DLSDU_length_ind_high
MS0	BBUB	NIL	Responder	0 to 244
MS0	BBUU-OM	NIL	Subscriber, Responder/Subscriber	244
MS0	BBUU-OM	NIL	Responder	0
MS0	BQUU	61	Responder	7 to 244
MS0	BQUU	62	Responder	1 to 244
MS0	BQUU	53	Responder	5 to 244
MS0	BBUU	60	Responder	0
MS0	QBCB	55	Responder	4 to 244
MS0	QBCB	56, 57, 59	Responder	0
MS0	QQUU-OM	58	Responder	2
MS1	QQCB-CO	50, 51	Responder	0
MS1	QQUU-CO	60	Responder	0
MS2	QQCB-CO	0 to 48	Responder	0
MS2	QQUU-CO	49	Responder	0
MS3	QQUU-OM	CS	Responder	0

CR Type = BBUU-OM & Role_in_Service = Initiator or Both
This constraint reflects the attributes for this CR if the value of the attribute CR Type is set to BBUU-OM and the attribute Role_in_Service is set to Subscriber or Responder/Subscriber.

Publisher Address
This dynamic attribute indicates the DL-address of a DP-slave (Publisher) the Subscriber shall subscribe from. The following rules shall apply for this attribute:

A value between 0 and 125 represents a valid Publisher address and therefore an active DXB-Link. The value 127 indicates that this DXB-Link Entry is no longer active.

This attribute shall be set to the value 127 with the Load CRL service. After reception of the DXB-Linktable with the Check User Prm or Check Ext User Prm service, the User shall set this attribute accordingly with the Load CRL DXB Link Entries service.

The MS0 AR state machine set shall set this attribute to the value 127 if the DP-slave acting as Subscriber leaves the data exchange mode.

Attribute Type: Unsigned8

Allowed values: 0 to 125, 127

Input Data Length Publisher
This static attribute indicates the number of octets of the Simple or Extended Input Data object of the related Publisher. This attribute shall only be evaluated by the MS0 AR state machine if the value of the attribute Publisher Address is not equal to 127.

This attribute shall be set with the Load CRL service. After reception of the DXB-Linktable with the Check User Prm or Check Ext User Prm service, the User shall set this attribute accordingly with the Load CRL DXB Link Entries service.

Attribute Type: Unsigned8

Allowed values: 1 to 244

Source Offset Publisher
This static attribute indicates the beginning (offset) of the subscription within the value of the Simple or Extended Input Data object of the related Publisher given with the attribute Publisher Address. This attribute shall only be evaluated by the MS0 AR state machine if the value of the attribute Publisher Address is not equal to 127.

This attribute shall be set with the Load CRL service. After reception of the DXB-Linktable with the Check User Prm or Check Ext User Prm service, the User shall set this attribute accordingly with the Load CRL DXB Link Entries service.

Attribute Type: Unsigned8

Allowed values: 0 to 243

Length
This static attribute indicates the number of octets the Subscriber shall subscribe from the related Publisher starting at the offset given in the attribute Source Offset Publisher. This attribute shall only be evaluated by the MS0 AR state machine if the value of the attribute Publisher Address is not equal to 127.

This attribute shall be set with the Load CRL service. After reception of the DXB-Linktable with the Check User Prm or Check Ext User Prm service, the User shall set this attribute accordingly with the Load CRL DXB Link Entries service.

Attribute Type: Unsigned8

Allowed values: 1 to 244

CR Name
This static attribute contains the name of the related CRL Entry. This attribute shall be set accordingly with the Load CRL service.

6.2.10.4.1.3 Invocation of CRL DP-slave object

For the invocation of the CRL DP-slave object the following rule has to be considered:

Only one CRL DP-slave object shall be invoked in a DP-slave.

6.2.10.4.2 CRL DP-master (class 1) class specification

6.2.10.4.2.1 Template

The CRL DP-master (Class 1) object is described by the following template:

DP ASE:			**AR ASE**
CLASS:			**CRL DP-master (Class 1)**
CLASS ID:			not used
PARENT CLASS:			TOP
ATTRIBUTES:			
1.	(m)	Key Attribute:	Implicit
2.	(m)	Attribute:	List of CRL Entries
2.1.	(m)	Attribute:	CREP
2.2.	(m)	Attribute:	CR Type
2.3.	(m)	Attribute:	Related AREP
2.4.	(m)	Attribute:	S_SAP_index
2.5.	(m)	Attribute:	D_SAP_index
2.6.	(m)	Attribute:	D_addr
2.7.	(m)	Attribute:	Service_activate
2.8.	(m)	Attribute:	Role_in_service
2.9.	(c)	Constraint:	Service_activate = SRD or MSRD
2.9.1.	(m)	Attribute:	Indication_mode
2.10.	(m)	Attribute:	Max_DLSDU_length_req_low
2.11.	(m)	Attribute:	Max_DLSDU_length_req_high
2.12.	(m)	Attribute:	Max_DLSDU_length_ind_low

2.13. (m) Attribute: Max_DLSDU_length_ind_high
2.14. (c) Constraint: CR Type = BQUU & D_SAP_index = 61
2.14.1. (m) Attribute: Active
2.14.2. (m) Attribute: New Prm
2.14.3. (m) Attribute: Prm Data
2.14.3.1. (m) Attribute: Lock
2.14.3.2. (m) Attribute: Sync
2.14.3.3. (m) Attribute: Freeze
2.14.3.4. (m) Attribute: WD On
2.14.3.5. (m) Attribute: WD Fact 1
2.14.3.6. (m) Attribute: WD Fact 2
2.14.3.7. (m) Attribute: Min Tsdr
2.14.3.8. (m) Attribute: Ident Number
2.14.3.9. (m) Attribute: Group Ident
2.14.3.10. (o) Attribute: DPV1 Enable
2.14.3.11. (o) Attribute: Fail Safe
2.14.3.12. (o) Attribute: Enable Publisher
2.14.3.13. (o) Attribute: WD Base
2.14.3.14. (o) Attribute: Enable Pull Plug Alarm
2.14.3.15. (o) Attribute: Enable Process Alarm
2.14.3.16. (o) Attribute: Enable Diagnosis Alarm
2.14.3.17. (o) Attribute: Enable Manufacturer Specific Alarm
2.14.3.18. (o) Attribute: Enable Status Alarm
2.14.3.19. (o) Attribute: Enable Update Alarm
2.14.3.20. (o) Attribute: Check Cfg Mode
2.14.3.21. (o) Attribute: Alarm Mode
2.14.3.22. (m) Attribute: Prm Structure
2.14.3.23. (c) Constraint: Prm Structure = FALSE
2.14.3.23.1 (m) Attribute: User Prm Data
2.14.3.23.1.1. (o) Attribute: Device Prm Data
2.14.3.23.1.2. (o) Attribute: Module Prm Data
2.14.3.24. (c) Constraint: Prm Structure = TRUE
2.14.3.24.1. (m) Attribute: List of Structured Prm Data
2.14.3.24.1.1. (o) Attribute: Structured User Prm Data
2.14.3.24.1.1.1. (o) Attribute: Device Related User Parameter
2.14.3.24.1.1.1.1. (m) Attribute: Device Prm Data
2.14.3.24.1.1.2. (o) Attribute: Module Related User Parameter
2.14.3.24.1.1.2.1. (m) Attribute: Slot Number
2.14.3.24.1.1.2.2. (m) Attribute: Module Prm Data
2.14.3.24.1.2. (o) Attribute: List of DXB Link Entries
2.14.3.24.1.2.1. (m) Attribute: Publisher Address
2.14.3.24.1.2.2. (m) Attribute: Input Data Length Publisher
2.14.3.24.1.2.3. (m) Attribute: Source Offset Publisher
2.14.3.24.1.2.4. (m) Attribute: Length
2.14.3.24.1.3. (o) Attribute: IsoM Parameter
2.14.3.24.1.3.1. (m) Attribute: T_{BASE_DP}
2.14.3.24.1.3.2. (m) Attribute: T_{DP}
2.14.3.24.1.3.3. (m) Attribute: T_{MAPC}
2.14.3.24.1.3.4. (m) Attribute: T_{BASE_IO}
2.14.3.24.1.3.5. (m) Attribute: T_{I}
2.14.3.24.1.3.6. (m) Attribute: T_{O}
2.14.3.24.1.3.7. (m) Attribute: T_{PLL_W}
2.14.3.24.1.3.8. (m) Attribute: T_{PLL_D}
2.14.3.24.1.4. (o) Attribute: List of DXB Subscriber Entries
2.14.3.24.1.4.1. (s) Attribute: Publisher Address
2.14.3.24.1.4.2.1. (m) Attribute: Input Data Length Publisher
2.14.3.24.1.4.2.2. (m) Attribute: Source Offset Publisher
2.14.3.24.1.4.3. (s) Attribute: Master Data
2.14.3.24.1.4.3.1. (m) Attribute: Source Offset Master
2.14.3.24.1.4.4. (m) Attribute: Dest Slot Number
2.14.3.24.1.4.5. (m) Attribute: Offset Data Area
2.14.3.24.1.4.6. (m) Attribute: Length Data Area
2.14.3.24.1.5. (o) Attribute: Prm Command
2.14.3.24.1.6. (c) Constraint: Prm Command supported = TRUE
2.14.3.24.1.6.1. (m) Attribute: Seq Number
2.14.3.24.1.6.2. (m) Attribute: Primary Request
2.14.3.24.1.6.3. (m) Attribute: MS1 Command

2.14.3.24.1.6.4. (m) Attribute: Master State Clear
2.14.3.24.1.6.5. (m) Attribute: Check Properties
2.14.3.24.1.6.5.1. (m) Attribute: Primary Request used
2.14.3.24.1.6.5.2. (m) Attribute: MS1 Command used
2.14.3.24.1.6.5.3. (m) Attribute: Address Change
2.14.3.24.1.6.5.4. (m) Attribute: Address Offset
2.14.3.24.1.6.6. (m) Attribute: Output Hold Time
2.14.3.24.1.7. (o) Attribute: Time AR Parameter
2.14.3.24.1.7.1. (m) Attribute: Clock Sync Interval
2.14.3.24.1.7.2. (m) Attribute: CS Delay Time
2.14.3.25. (o) Attribute: IsoM Req
2.15. (c) Constraint: CR Type = BQUU & D_SAP_index = 62
2.15.1. (m) Attribute: List of Cfg Data
2.15.1.1. (s) Attribute: Cfg Identifier
2.15.1.1.1. (m) Attribute: Type
2.15.1.1.2. (m) Attribute: Format
2.15.1.1.3. (m) Attribute: Consistency
2.15.1.1.4. (m) Attribute: Length
2.15.1.2. (s) Attribute: Special Cfg Identifier
2.15.1.2.1. (m) Attribute: Type
2.15.1.2.2. (c) Constraint: Type = Input or Input-Output
2.15.1.2.2.1. (m) Attribute: Input Length
2.15.1.2.2.2. (m) Attribute: Input Format
2.15.1.2.2.3. (m) Attribute: Input Consistency
2.15.1.2.3. (c) Constraint: Type = Output or Input-Output
2.15.1.2.3.1. (m) Attribute: Output Length
2.15.1.2.3.2. (m) Attribute: Output Format
2.15.1.2.3.3. (m) Attribute: Output Consistency
2.15.1.2.4. (o) Attribute: List of Data Description
2.15.1.2.4.1. (m) Attribute: Data Description
2.15.1.2.5. (o) Attribute: Manufacturer Specific Data
2.16. (c) Constraint: CR Type = BBUU & D_SAP_index = 60
2.16.1. (m) Attribute: Diag Upd Delay
2.16.2. (m) Attribute: Ext Diag Overflow
2.16.3. (m) Attribute: Deactivated
2.16.4. (m) Attribute: Station Non Existent
2.16.5. (m) Attribute: Invalid Slave Response
2.17. (c) Constraint: CR Type = BBUB & D_SAP_index = NIL
2.17.1. (m) Attribute: Fail Safe
2.17.2. (m) Attribute: DPV1 Supported
2.17.3. (m) Attribute: NA to Abort
2.17.4. (m) Attribute: Ignore AClr
2.17.5. (m) Attribute: Output Len
2.17.6. (m) Attribute: Exp Inp Len
2.17.7. (m) Attribute: Data Transfer Status
2.17.8. (m) Attribute: System Diagnosis Status
2.17.9. (m) Attribute: Publisher Flag
2.18. (c) Constraint: CR Type = BQUU & D_SAP_index = 53
2.18.1. (m) Attribute: List of Structured Prm Data
2.18.1.1. (o) Attribute: Structured User Prm Data
2.18.1.1.1. (o) Attribute: Device Related User Parameter
2.18.1.1.1.1. (m) Attribute: Device Prm Data
2.18.1.1.2. (o) Attribute: Module Related User Parameter
2.18.1.1.2.1. (m) Attribute: Slot Number
2.18.1.1.2.2. (m) Attribute: Module Prm Data
2.18.1.2. (o) Attribute: List of DXB Link Entries
2.18.1.2.1. (m) Attribute: Publisher Address
2.18.1.2.2. (m) Attribute: Input Data Length Publisher
2.18.1.2.3. (m) Attribute: Source Offset Publisher
2.18.1.2.4. (m) Attribute: Length
2.18.1.3. (o) Attribute: IsoM Parameter
2.18.1.3.1. (m) Attribute: T_{BASE_DP}
2.18.1.3.2. (m) Attribute: T_{DP}
2.18.1.3.3. (m) Attribute: T_{MAPC}
2.18.1.3.4. (m) Attribute: T_{BASE_IO}
2.18.1.3.5. (m) Attribute: T_I
2.18.1.3.6. (m) Attribute: T_O

2.18.1.3.7. (m) Attribute: T_{PLL_W}
2.18.1.3.8. (m) Attribute: T_{PLL_D}
2.18.1.4. (o) Attribute: List of DXB Subscriber Entries
2.18.1.4.1. (s) Attribute: Publisher Address
2.18.1.4.2.1. (m) Attribute: Input Data Length Publisher
2.18.1.4.2.2. (m) Attribute: Source Offset Publisher
2.18.1.4.3. (s) Attribute: Master Data
2.18.1.4.3.1. (m) Attribute: Source Offset Master
2.18.1.4.4. (m) Attribute: Dest Slot Number
2.18.1.4.5. (m) Attribute: Offset Data Area
2.18.1.4.6. (m) Attribute: Length Data Area
2.18.1.5. (o) Attribute: Time AR Parameter
2.18.1.5.1. (s) Attribute: Clock Sync Interval
2.18.1.5.2. (s) Attribute: CS Delay Time
2.19. (o) Attribute: CR Name

SERVICES:

1. (m) MgtService: Load CRL
2. (o) MgtService: Get CRL
3. (m) MgtService: CRL Slave Activate
4. (m) MgtService: CRL Slave New Prm
5. (m) MgtService: CRL Slave New Prm Data

6.2.10.4.2.2 Attributes

Implicit
The attribute Implicit indicates that the CRL DP-master (Class 1) object is implicitly addressed by the service.

List of CRL Entries
This attribute is composed of the following list elements:

CREP
This static attribute contains the local identifier of this CR. This attribute shall be set accordingly with the Load CRL service.

CR Type
This static attribute indicates the type of this CR. This attribute is dependent on the AR Type of the related AR. This attribute shall be set accordingly with the Load CRL service. For each AR Type the correspondent CRL entries shall be set.

Table 222 shows the allowed values for this attribute.

Related AREP
This static attribute contains the local identifier for the related AR for this CR. This attribute shall be set accordingly with the Load CRL service.

S_SAP_index
This static attribute contains the identifier for the related Service Access Point of the local Data Link Layer. The value for this attribute has to be set correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR. This attribute shall be set accordingly with the Load CRL service.

Table 222 shows the allowed values for this attribute.

Table 222 – S_SAP_index

AR Type	CR Type	S_SAP_index	Remark
MS0	BBUB	NIL	Mandatory for the MS0 AR
MS0	BQUU	62	Mandatory for the MS0 AR
MS0	BBUU	62	Mandatory for the MS0 AR
MS0	QQUU-OM	62	Mandatory for the MS0 AR
MS1	QQCB-CO	51	Mandatory for the MS1 AR
MS1	QQUU-OO	62	Mandatory for the MS1 AR
MS3	QQUU-OM	CS	Mandatory for the MS3 AR
MM1	QQCB-CL	54	Mandatory for the MM1 AR
MM2	QQUU-OM	54	Mandatory for the MM2 AR

D_SAP_index

This static attribute contains the identifier for the related Service Access Point of the remote Data Link Layer. The value for this attribute has to be set correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR. This attribute shall be set accordingly with the Load CRL service. Table 223 shows the allowed values for this attribute.

Table 223 – D_SAP_index

AR Type	CR Type	D_SAP_index	Remark
MS0	BBUB	NIL	Mandatory for the MS0 AR
MS0	BQUU	61, 62	Mandatory for the MS0 AR
MS0	BQUU	53	Optional for the MS0 AR
MS0	BBUU	60	Mandatory for the MS0 AR
MS0	QQUU-OM	58	Mandatory for the MS0 AR
MS1	QQCB-CO	50	Optional for each MS1 AR
MS1	QQCB-CO	51	Mandatory for the MS1 AR
MS1	QQUU-OO	60	Mandatory for the MS1 AR
MS3	QQUU-OM	CS	Mandatory for the MS3 AR
MM1	QQCB-CL	54	Mandatory for the MM1 AR
MM2	QQUU-OM	54	Mandatory for the MM2 AR

D_addr

This dynamic attribute contains the DL-address of the remote communication partner. The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR. Table 224 shows the allowed values for this attribute used with the Load CRL service. Furthermore this attribute is influenced by the AR state machine dependent of the attribute CR Type of this CR and of the attribute AR Type of the related AR. Table 224 shows the allowed values for this attribute for the modification of the AR sate machine.

Table 224 – D_addr

AR Type	CR Type	S_SAP_index	Time Device Type (ARL)	D_addr as loaded with the Load CRL service	D_addr modified by the AR state machine
MS0	BBUB	NIL	--	0 to 126	-
MS0	BQUU	62	--	0 to 126	-
MS0	BBUU	62	--	0 to 126	-
MS0	QQUU-OM	62	--	0 to 127	-
MS1	QQCB-CO	51	--	0 to 126	-
MS1	QQUU-CO	62	--	0 to 126	-
MS3	QQUU-OM	CS	Time Master	127	-
MS3	QQUU-OM	CS	Time Receiver	None	All
MM1	QQCB-CL	54	--	All	0 to 125
MM2	QQUU-OM	54	--	All	0 to 125

Where

All	means that the local Service Access Point shall be activated with no access protection
None	means that the local Service Access Point shall not be activated
0 to 127	means that the attribute D_addr of the local Service Access Point shall be set with the corresponding remote DL-address
-	means that the AR state machines shall not modify the attribute D_addr
--	means that the value of this attribute is not relevant

Service_activate
This static attribute indicates the DL services which shall be activated for the S_SAP_index of this CRL entry.

The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR.

Table 225 shows the allowed values for this attribute.

Table 225 – Service_activate

AR Type	CR Type	S_SAP_index	Publisher Flag	Service_activate
MS0	BBUB	NIL	FALSE	SRD
MS0	BBUB	NIL	TRUE	MSRD
MS0	BQUU	62	-	SRD
MS0	BBUU	62	-	SRD
MS0	QQUU-OM	62	-	SDN
MS1	QQCB-CO	51	-	SRD
MS1	QQUU-CO	62	-	SRD
MS3	QQUU-OM	CS	-	CS Time Event CS Clock Value
MM1	QQCB-CL	54	-	SRD
MM2	QQUU-OM	54	-	SDN

Role_in_service
This static attribute indicates the role which shall be activated for the S_SAP_index of this CRL entry. The following values are specified:

- Initiator: The station initiates the respective service exclusively

Responder: The station response to the service exclusively

The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR.

Table 226 shows the allowed values for this attribute.

Table 226 – Role_in_service

AR Type	CR Type	S_SAP_index	Time Device Type (ARL)	Role_in_service
MS0	BBUB	NIL	-	Initiator
MS0	BQUU	62	-	Initiator
MS0	BBUU	62	-	Initiator
MS0	QQUU-OM	62	-	Initiator
MS1	QQCB-CO	51	-	Initiator
MS1	QQUU-CO	62	-	Initiator
MS3	QQUU-OM	CS	Time Master	Initiator
MS3	QQUU-OM	CS	Time Receiver	responder
MM1	QQCB-CL	54	-	Responder
MM2	QQUU-OM	54	-	Responder

Indication_mode

This conditional and static attribute shall only be present if the attribute Service_activate of this CRL entry has the value SRD or MSRD and if the attribute Role_in_service of this CRL entry has the value Responder. It indicates the Indication_mode which the DLL shall use to convey the indications of the (M)SRD service. The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type of this CR and to the attribute AR Type of the related AR.

Table 227 shows the allowed values for this attribute.

Table 227 – Indication_mode

AR Type	CR Type	S_SAP_index	Indication_mode
MS0	BBUB	NIL	-
MS0	BQUU	62	-
MS0	BBUU	62	-
MS0	QQUU-OM	62	-
MS1	QQCB-CO	51	-
MS1	QQUU-CO	62	-
MM1	QQCB-CL	54	Data
MM2	QQUU-OM	54	-

where

- means that the attribute Indication_mode is not present for this CR Type

Max_DLSDU_length_req_low

This static attribute indicates the maximum length for the DLSDU with low priority as requester/responder. The value for this attribute has to be set with the Load CRL service correspondent to the attributes CR Type and D_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 228 shows the allowed values for this attribute.

Table 228 – Max_DLSDU_length_req_low

AR Type	CR Type	D_SAP_index	Time Device Type (ARL)	Max_DLSDU_length_req_low
MS0	BBUB	NIL	-	0
MS0	BQUU	61, 62, 53	-	0
MS0	BBUU	60	-	0
MS0	QQUU-OM	58	-	0
MS1	QQCB-CO	50	-	4
MS1	QQCB-CO	51	-	4 to 244
MS1	QQUU-OO	60	-	0
MS3	QQUU-OM	CS	Time Master	17
MS3	QQUU-OM	CS	Time Receiver	0
MM1	QQCB-CL	54		4 to 244
MM2	QQUU-OM	54		0

Max_DLSDU_length_req_high
This static attribute indicates the maximum length for the DLSDU with high priority as requester/responder. The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type and D_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 229 shows the allowed values for this attribute.

Table 229 – Max_DLSDU_length_req_high

AR Type	CR Type	D_SAP_index	Max_DLSDU_length_req_high
MS0	BBUB	NIL	0 to 244
MS0	BQUU	61	7 to 244
MS0	BQUU	62	1 to 244
MS0	BQUU	53	5 to 244
MS0	BBUU	60	0
MS0	QQUU-OM	58	2
MS1	QQCB-CO	50, 51	0
MS1	QQUU-OO	60	0
MS3	QQUU-OM	CS	0
MM1	QQCB-CL	54	0
MM2	QQUU-OM	54	0

Max_DLSDU_length_ind_low
This static attribute indicates the maximum length for the DLSDU with low priority for indication. The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type and D_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 230 shows the allowed values for this attribute.

Table 230 – Max_DLSDU_length_ind_low

AR Type	CR Type	D_SAP_index	Time Device Type (ARL)	Max_DLSDU_length_ind_low
MS0	BBUB	NIL	-	0 to 244
MS0	BQUU	61,62, 53	-	0
MS0	BBUU	60	-	6 to 244
MS0	QQUU-OM	58	-	0
MS1	QQCB-CO	50	-	4
MS1	QQCB-CO	51	-	4 to 244
MS1	QQUU-OO	60	-	6 to 244
MS3	QQUU-OM	CS	Time Master	0
MS3	QQUU-OM	CS	Time Receiver	17
MM1	QQCB-CL	54		4 to 244
MM2	QQUU-OM	54		2

Max_DLSDU_length_ind_high
This static attribute indicates the maximum length for the DLSDU with high priority for indication. The value for this attribute has to be set with the Load CRL service correspondent to the attribute CR Type and D_SAP_index of this CR and to the attribute AR Type of the related AR.

Table 231 shows the allowed values for this attribute.

Table 231 – Max_DLSDU_length_ind_high

AR Type	CR Type	D_SAP_index	Max_DLSDU_length_ind_high
MS0	BBUB	NIL	0 to 244
MS0	BQUU	61,62, 53	0
MS0	BBUU	60	0
MS0	QQUU-OM	58	0
MS1	QQCB-CO	50, 51	0
MS1	QQUU-OO	60	0
MS3	QQUU-OM	CS	0
MM1	QQCB-CL	54	0
MM2	QQUU-OM	54	0

CR Type = BQUU & D_SAP_index = 61
This constraint reflects the attributes for this CR if the value of the attribute CR Type is set to BQUU and the attribute D_SAP_index is set to 61.

Active
This dynamic attribute indicates the state of interactions to the correspondent DP-slave. The attribute set to FALSE indicates that the DP-master (Class 1) shall stop the interactions to the correspondent DP-slave. In this state it is allowed to change the CRL entries of this DP-slave with the Load CRL service. If this attribute is set to TRUE the DP-master (Class 1) shall start the interactions to the correspondent DP-slave. In this state only the attribute Prm Data may be changed with the CRL Slave New Prm Data service (see also the attribute New Prm). This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave Activate service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

New Prm
This dynamic attribute indicates to the MS0 AR state machine to send new parameterization data during the data exchange mode to the correspondent DP-slave. The following rules shall be applied:

If the attribute is set to FALSE the DP-master (Class 1) transfers the input, output data to the correspondent DP-slave. If this attribute is set to TRUE and the attribute DPV1 Supported of the related CRL entry has the value FALSE the DP-master (Class 1) transfers for one DP-slave poll cycle new parameterisation data to the correspondent DP-slave. The MS0 AR state machine sets the attribute to FALSE if the transfer was successful.

This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm service. This attribute is further influenced by the MS0 AR state machine.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Prm Data
This dynamic attribute is an aggregate of the parameterization data of the correspondent DP-slave and is composed of the following elements. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Lock
This dynamic attribute is used by the MS0 AR state machine to assign or release the correspondent DP-slave. This attribute is set to LOCK if the attribute Active of this related CRL entry is set to TRUE. This attribute is set to UNLOCK if the attribute Active of this related CRL entry is set to FALSE, or due to failures during the data exchange mode. This attribute shall be set to LOCK with the Load CRL service. This attribute is further influenced by the MS0 AR state machine. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Allowed values: LOCK, UNLOCK

Sync
This dynamic attribute indicates to the correspondent DP-slave that it shall participate in the synchronization of the Output Data (Sync mode). The DP-slave shall check whether it supports the requested functionality and if the DP-slave does not support this function it shall respond with an error entry in the Slave Diagnosis (Not Supported). This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Table 232 shows the allowed values for this attribute.

Table 232 – Sync

Value	Meaning
FALSE	Sync ability not requested for this DP-slave
TRUE	Sync ability requested for this DP-slave

Freeze
This dynamic attribute indicates to the correspondent DP-slave that it shall participate in the synchronization of the Input Data (Freeze mode). The DP-slave shall check whether it supports the requested functionality and if the DP-slave does not support this function it shall respond with an error entry in the Slave Diagnosis

(Not Supported). This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Table 233 shows the allowed values for this attribute.

Table 233 – Freeze

Value	Meaning
FALSE	Freeze ability not requested for this DP-slave
TRUE	Freeze ability requested for this DP-slave

WD On
This dynamic attribute indicates if the watchdog control shall be activated at the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

WD Fact 1, WD Fact 2
These dynamic attributes contain the values for the monitoring of the DP-master (Class 1) at the MS0 AR (watchdog control (T_{WD})) at the correspondent DP-slave. The watchdog control in a DP-slave takes care that, if the DP-master fails, the outputs will be set in the safe state after the expiration of this time. The time is calculated according to equation:

T_{WD} = Watchdog Time Base × WD Fact 1 × WD Fact 2

If the attribute DPV1 Supported has the value FALSE the Watchdog Time Base shall be set to 10 ms.

If the attribute DPV1 Supported has the value TRUE the Watchdog Time Base shall be set to the value 1ms if the attribute WD Base has the value TRUE. If the attribute DPV1 Supported has the value TRUE the Watchdog Time Base shall be set to the value 10ms if the attribute WD Base has the value FALSE. These attributes shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Allowed values: 1 to 255

Min TSDR
This dynamic attribute contains the value for the Minimum Station Delay Responder time of the DLL of the correspondent DP-slave. If this attribute is equal zero, the previous value of the Min TSDR remains unchanged. These attributes shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Allowed values: 0 to 255

Ident Number
This dynamic attribute contains the Ident Number of the correspondent DP-slave. The DP-slave accepts the parameterization only if the transmitted Ident Number is identical with the attribute Ident Number of the MS0 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Attribute Type: Unsigned16

Allowed values: 0 to 2^{16} - 1

Group Ident
This dynamic attribute indicates the group(s) to which the correspondent DP-slave shall be assigned by the DP-master (Class 1) in the parameterization. The Group Ident distinguishes 8 groups. If the Group Ident is set to No Group, the DP-slave is not assigned to any group. The DP-slave stores the value of this attribute in the attribute Group Identifier of the MS0 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Allowed values: Group 1 to Group 8, No Group and combinations

DPV1 Enabled
This optional and static indicates if the extended functionality shall be activated at the correspondent DP-slave. This attribute shall only be present if the attribute DPV1 Supported in the related MS0 AR entry in the List of ARL Entries of the DP-slave has the value TRUE. This attribute shall be set accordingly with the Load CRL service.

Table 234 shows the allowed values for this attribute.

Table 234 – DPV1 enabled

Value	Meaning
FALSE	The extended functionality related with the attribute DPV1 Supported is disabled.
TRUE	The extended functionality related with the attribute DPV1 Supported is enabled.

Fail Safe
This optional and static attribute indicates to the correspondent DP-slave if the DP-master (Class 1) supports the Fail Safe mode. If the DP-slave requires the Fail Safe mode and the DP-master (Class 1) does not support this mode the DP-slave shall respond with an error entry in the Slave Diagnosis (Prm Fault). The DP-slave stores the value of this attribute in the attribute Fail Safe of the MS0 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Table 235 shows the allowed values for this attribute.

Table 235 – Fail safe

Value	Meaning
FALSE	The DP-master (Class 1) does not support the Fail Safe mode.
TRUE	The DP-master (Class 1) supports the Fail Safe mode.

Enable Publisher
This optional and static attribute indicates to the correspondent DP-slave that it shall operate as Publisher. The DP-slave shall check whether it supports the requested functionality. If the DP-slave does not support this function it shall respond with an appropriate entry in the Diagnosis information (Prm Fault). This attribute shall be set accordingly with the Load CRL service.

Table 236 shows the allowed values for this attribute.

Table 236 – Enable publisher

Value	Meaning
FALSE	Publisher ability not requested for this DP-slave
TRUE	Publisher ability requested for this DP-slave

WD Base
This optional and static attribute contains the Time Base value for the watchdog control according to the parameterization for the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute WD Base of the MS0 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Table 237 shows the allowed values for this attribute.

Table 237 – WD base

Value	Meaning
FALSE	The Watchdog Time Base is 10ms.
TRUE	The Watchdog Time Base is 1ms.

Enable Pull Plug Alarm
This optional and static attribute indicates if the transmission of the Pull Plug Alarm shall be enabled at the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute Pull Plug Alarm enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Allowed values: TRUE, FALSE

Enable Process Alarm
This optional and static attribute indicates if the transmission of the Process Alarm shall be enabled at the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute Process Alarm enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Enable Diagnosis Alarm
This optional and static attribute indicates if the transmission of the Diagnosis Alarm shall be enabled at the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute Diagnosis Alarm enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Enable Manufacturer Specific Alarm
This optional and static attribute indicates if the transmission of the Manufacturer Specific Alarm shall be enabled at the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute Manufacturer Specific Alarm enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Enable Status Alarm
This optional and static attribute indicates if the transmission of the Status Alarm shall be enabled at the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute Status Alarm enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Enable Update Alarm
This optional and static attribute indicates if the transmission of the Update Alarm shall be enabled at the correspondent DP-slave. The DP-slave stores the value of this attribute in the attribute Update Alarm enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Check Cfg Mode
This optional and static attribute indicates how the validation of the Cfg Data shall be performed in the correspondent DP-slave. The value of this attribute is passed to the application of the DP-slave with the Check Cfg service and influences the check of the Cfg Data (see 6.2.5.3.5). This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Alarm Mode
This optional and static attribute indicates the maximum number of possible active alarms between the DP-master (Class 1) and this DP-slave. The DP-slave stores the value of this attribute in the attribute Alarm Mode enabled of the MS1 AR entry in the List of ARL Entries of the DP-slave. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Table 238 shows the allowed values for this attribute.

Table 238 – Alarm mode

Value	Meaning
0	1 alarm of each type
1	2 alarms in total
2	4 alarms in total
3	8 alarms in total
4	12 alarms in total
5	16 alarms in total
6	24 alarms in total
7	32 alarms in total

Prm Structure
This static attribute indicates which type of parameterization objects shall be provided by the Check User Prm service at the correspondent DP-slave. The value of this attribute is passed to the application of the DP-slave with the Check User Prm service. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Boolean

Allowed values: TRUE, FALSE

Prm Structure = FALSE
This constraint reflects that the parameter which follow are only related to the MS0 User Parameter object.

User Prm Data
This dynamic attribute contains the value(s) for the MS0 User Parameter object (either Device Related User Parameter or a List of Module Related User Parameter or both) of the correspondent DP-slave and is composed of the following elements. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Device Prm Data
This optional and dynamic attribute contains the value for the Device Related User Parameter of the MS0 User Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Module Prm Data
This optional and dynamic attribute contains the value for the Module Related User Parameter of the MS0 User Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service. This attribute may be set dynamically by the CRL Slave New Prm Data service.

Prm Structure = TRUE
This constraint reflects that the parameter which follow are related to the MS0 Structured User Parameter object and/or the DXB-Linktable object and/or the IsoM Parameter object.

List of Structured Prm Data
This dynamic attribute is composed of one or more of the following elements.

Structured User Prm Data
This optional and dynamic attribute is composed of one or both of the following elements.

Device Related User Parameter
This static attribute indicates the presence of Device Related User Parameter. This attribute shall be set accordingly with the Load CRL service.

Device Prm Data
This static attribute contains the value for the Device Related User Parameter of the MS0 Structured User Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Module Related User Parameter
This static attribute indicates the presence of Module Related User Parameter. This attribute shall be set accordingly with the Load CRL service.

Slot Number
This static attribute contains the value for the attribute Slot Number of the MS0 Structured User Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Module Prm Data
This static attribute contains the value for the attribute Module Prm Data of the MS0 Structured User Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

List of DXB Link Entries
This optional and static attribute indicates the presence of the DXB-Linktable Entries and is composed of the following elements. This attribute shall be set accordingly with the Load CRL service.

Publisher Address
This static attribute contains the value for the attribute Publisher Address of one DXB-Linktable entry of the DXB-Linktable object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned8

Allowed values: 0 to 125

Input Data Length Publisher
This static attribute contains the value for the attribute Input Data Length Publisher of one DXB-Linktable entry of the DXB-Linktable object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Allowed values: 1 to 244

Source Offset Publisher
This static attribute contains the value for the attribute Source Offset Publisher of one DXB-Linktable entry of the DXB-Linktable object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned8

Allowed values: 0 to 243

Length
This static attribute contains the value for the attribute Length of one DXB-Linktable entry of the DXB-Linktable object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned8

Allowed values: 1 to 244

IsoM Parameter
This optional and static attribute indicates the presence of the IsoM Parameter and is composed of the following elements. This attribute shall be set accordingly with the Load CRL service.

T_{BASE_DP}
This static attribute contains the value for the attribute T_{BASE_DP} of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Allowed values: 375, 750, 1 500, 3 000, 6 000, 12 000 (the value 1 500 is mandatory for standard Isochronous Mode)

T_{DP}
This static attribute contains the value for the attribute T_{DP} of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned16

Allowed values: 1 to 2^{16} - 1

T_{MAPC}
This static attribute contains the value for the attribute T_{MAPC} of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Allowed values: 1 to 14

T_{BASE_IO}
This static attribute contains the value for the attribute T_{BASE_IO} of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Allowed values: 375, 750, 1 500, 3 000, 6 000, 12 000 (the value 1 500 is mandatory for standard Isochronous Mode)

T_I
This static attribute contains the value for the attribute T_I of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned16

Allowed values: 0 (special case), 1 to 2^{16} - 1

T_O
This static attribute contains the value for the attribute T_O of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned16

Allowed values: 0 (special case), 1 to 2^{16} - 1

T_{DX}
This static attribute contains the value for the attribute T_{DX} of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned32

Allowed values: 1 to 2^{32} - 1

T_{PLL_W}
This static attribute contains the value for the attribute T_{PLL_W} of the IsoM Parameter object of the correspondent DP-slave. This attribute shall be set accordingly with the Load CRL service.

Attribute Type: Unsigned16