P3 Companion Standard

Dutch Smart Meter Requirements

By order of:	**Enbin**
Reference:	**B1043**
Date:	**April 18th, 2008**
Version:	**2.2**
Status:	**Final**

Change Summary

Revision	Created / Modified	Date	Approved	Comment
1.00	H. Spelt	28-June-07	WG04	Original release
1.20	H. Spelt	12-Sep-07	WG04	NTA first draft
2.0 first draft	H. Spelt	27-Nov-07	WG04	Table of Contents modified + security and protocol stacks added; introduction prepared
2.0 second draft	H. Spelt S. Ermens	13-Dec-07	WG04	Excel file integrated, new classes added in appendix, review comments DLMS meeting 4 December added, integrated Event and error handling document
2.0 final draft	H. Spelt S. Ermens	21-Dec-07	WG04	- Update of Disconnector class (Appendix A3); - New OBIS codes added and Class id's. (excel 13/12/07); - Review comments WG04 members added; - Security part added; - update event and error handling v0.3 (table event codes and error codes updated); - Latest excel file object mapping dated 20/12/07 consistency check with CS P3 done
2.1 final	H. Spelt	4-Feb-08	WG04	- Page 46 highlight reference added - OBIS codes changed for Limiter, Disconnect control and control log - Setup objects appended with attributes and OBIS codes - Selection of security suite 7 mentioned - References added/modified - Yellow markings updated/removed/added - Push/pull clarification added - Footnote 3 and 4 added - Rate 1 and 2 mapped on Low and Normal Tariff added as extra information - PLC objects added in Appendix B
2.11 draft	H. Spelt	12-Feb-08	WG04	- Update of Event and error handling version 1.0 added (see chapter 4) - 'Active import (-A)' changed to 'Active export (-A)' in chapter 6.2

2.13 draft	H. Spelt	16-Mar-08	WG04	- Review remarks updated after the 21st February 2008 DLMS meeting with vendors - Including results of outstanding action points (event handling => 3 comments August for event codes added, - Chapter 4.2.5 OBIS codes; OBIS for event codes corrected) - Chapters 2.1 and 2.2 added - TCP-UDP setup => added port no. (Default 4059) - IPv4 setup modified some parameters - PPP setup modifies some parameters - Ethernet setup parameter MAC_address modified - Comment added by August to review "Message security" paragraph
2.2 final	H. Spelt	18-Apr-08	WG04	- Review remarks updated after the 19th March 2008 DLMS meeting with vendors - Chapters 2.1 and 2.2 were modified (conformance blocks replaced by services supported) - "Message security" paragraph rewritten - Firmware E upgrade description added (par. 5.13) - Two event codes (17,18) added of firmware upgrade - Some binding information was added in chapter 7.2 (M-bus devices) - Appendix A and B (new classes) was updated with latest modifications of DLMS UA - Added single action scheduler for firmware activation with OBIS code

Distribution information

Version	Addressees	Remarks
1.0	Members WG4	For Review
1.20	Members WG4	For Review
2.0 first draft	Members WG4 Manufacturers	For Review
2.0 second draft	Members WG4 Manufacturers	For Review
2.0 final draft	Members WG4 Manufacturers	For Review
2.1 final	Members WG4 Manufacturers Grid Operators	For Review
2.11 draft	Members WG04	For Review
2.13 draft	Members WG04	For Review
2.2 final	Members WG4 Manufacturers Grid Operators	For Review

Issue list

Version	Activity
2.0	A few yellow markings/comments are (still) not clarified (especially chapter 7.2)
2.1	GPRS way of connection handling/setup (like push mechanism) not yet added
2.2	Add requirements for Ethernet communication, if necessary.

CONTENTS

1 INTRODUCTION

1.1 Scope

This document provides a companion standard for an Automatic Meter Reading (AMR) system for electricity thermal, (heat & cold), gas and water meters.

The scope of this standard is on:

- Residential electricity meters
- Residential thermal (heat & cold) meters
- Residential gas meters and gas valve
- Residential water meters

This companion standard focuses on the P3 interface for Electricity meters.

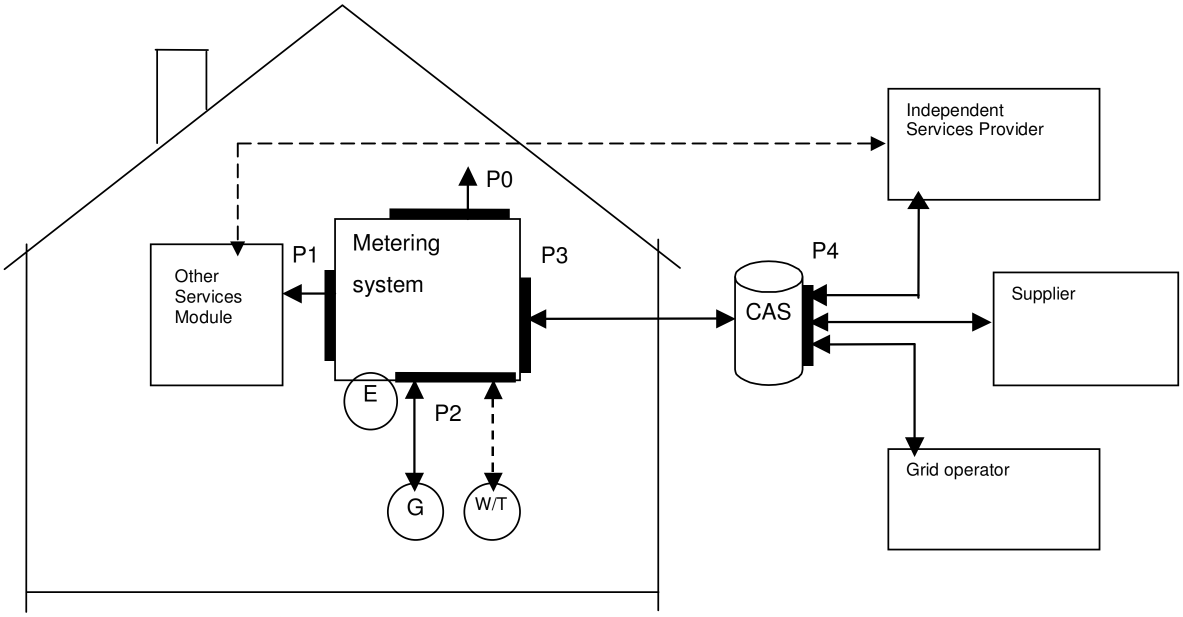


Figure 1.1: Meter interfaces overview

The goal of this companion standard is to reach an open, standardized protocol implementation based on DLMS/COSEM.

This companion standard is the result of a combined effort of the major Dutch grid operators and different manufactures who defined the necessary DLMS/COSEM object mapping.

1.2 System architecture

The communication interface P3 and P3.1 (see figure 1.2) is based on the DLMS/COSEM standard. References to the DLMS/COSEM standard are included in section 1.3. This companion standard P3 only includes deviations, clarifications or additions to the standard as defined in the relevant standard documents. P3.2 and P0 interface (see also figure 1.2) are not part of this companion standard. The P1 and P2 interfaces are described in separate companion standards.

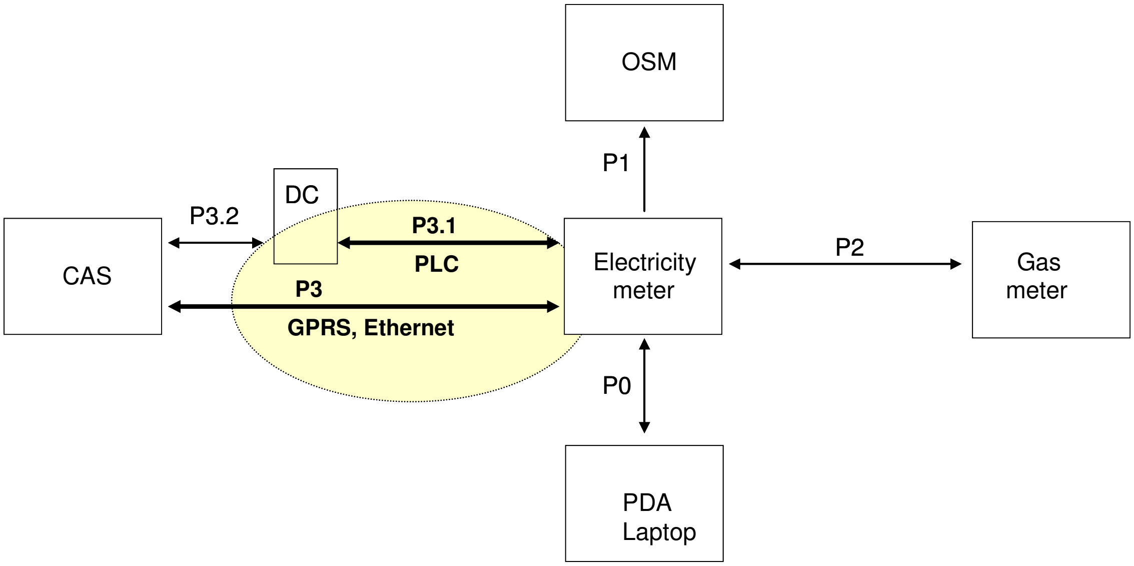


Figure 1.2: DLMS/COSEM infrastructure P3

1.3 Normative references

The following standards are referred to in this companion standard. For undated references the latest edition applies.

Ref No	Document	Description
1	DLMS UA 1000-1 ed.8, 2007[1]	Blue book, COSEM Identification System and Interface Classes
2	DLMS UA 1000-2 ed.6, 2007[2]	Green book, DLMS/COSEM Architecture and Protocols
3	DLMS UA 1001-1 ed.3, 2007	Yellow book, DLMS/COSEM Conformance Testing Process
4	DLMS UA 1002: ed.1, 2003	White book, COSEM Glossary of Terms

[1] A new version (edition 9) of the Blue book is in development. The new version of the blue book will describe the new Classes and OBIS codes mentioned in this Companion standard and gives also the description of the new encryption and authentication mechanism.

[2] A newer version (edition 7) of the Green book will have the definition of PLC Communication Profile and also the description of the new encryption and authentication mechanism.

Ref No	Document	Description
5	IEC 61334-4-32	Distribution automation using distribution line carrier systems – Part 4: Data communication protocols – Section 32: Data link layer – Logical link control (LLC)
6	IEC 61334-4-511	Distribution automation using distribution line carrier systems – Part 4-511: Data communication protocols – System management : CIASE protocol
7	IEC 61334-4-512	Distribution automation using distribution line carrier systems – Part 4-512: Data communication protocols – System management using profile 61334-5-1 Management Information Bases (MIB)
8	IEC 61334-5-1	Distribution automation using distribution line carrier systems – Part 5-1: Lower layer profiles – The spread frequency shift keying (S-FSK) profile
9	IEC 62056-21 Ed 1.0:2002	Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange
10	IEC 62056-42 Ed.1.0:2002	Electricity metering – Data exchange for meter reading, tariff and load control – Part 42: Physical layer services and procedures for connection-oriented asynchronous data exchange
11	IEC 62056-46 Ed.1.1:2007	Electricity metering – Data exchange for meter reading, tariff and load control – Part 46: Data link layer using HDLC protocol
12	IEC 62056-47 Ed 1.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 47: COSEM transport layer for IP networks
13	IEC 62056-53 Ed 2.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 53: COSEM Application layer
14	IEC 62056-61 Ed 2.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: OBIS Object identification system
15	IEC 62056-62 Ed 2.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 62: Interface classes
16	NTA 8130 NL:2007	*Basisfuncties voor de meetinrichting voor elektriciteit, gas en thermische energie voor kleinverbruikers*
17	ISO/IEC 8802.2	Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 2: Logical Link Control -Description
18	ISO/IEC 13239	Information technology -- Telecommunications and information exchange between systems -- High-level data link control (HDLC) procedures
19	EN 13757-2	Communication systems for and remote reading of meters – Part 2: Physical and link layer
20	EN 13757-3 : 2004	Communication systems for and remote reading of meters – Part 3: Dedicated application layer
21	B101: Smart Meter Requirements	Extended Dutch Smart Meter specification
22	Cosem security	WG04 interoperability & Communications (Milan Kozole); December 2007
23	Project: DLMS/COSEM over PLC	COSEM interface classes for setting up and managing communication profiles using power line carrier (latest update of Gyozo Kmethy dated 29 February 2008) and Extension of the COSEM object model for smart metering ((latest update of Gyozo Kmethy dated 3 March 2008)

Remark:

The existing IEC 62056-series do not describe and cover yet all functionality of DLMS/COSEM required by this companion standard P3. The standardization process of this new functionality will start when the new Green book (edition 7) and Blue book (edition 9) are approved/published by the DLMS User Association.

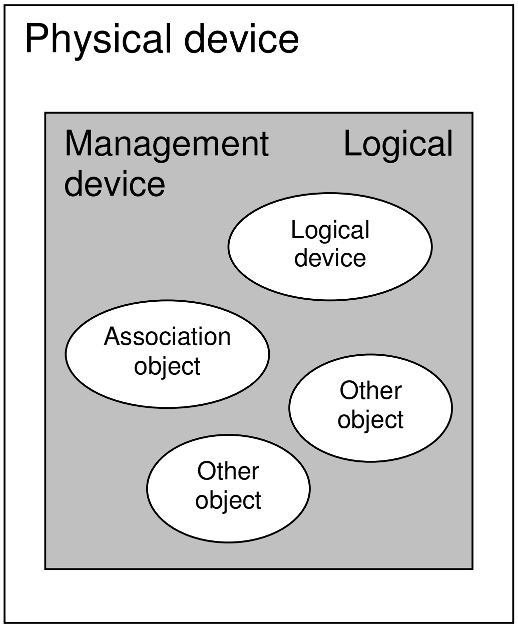
1.4 Document list

Following table shows the complete set of documents that build up the Dutch Smart Meter Requirements, of which this tender document is a part of.

#	Document name postfix	description
[1]	Main	The main document of the Dutch Smart Meter Requirements, containing all definitions and most of the use cases and requirements
[2]	Tender	Tender document, containing additional general requirements, use cases and performance requirements
[3]	P1	Companion standard P1
[4]	P2	Companion standard P2
[5]	P3	Companion standard P3

2 LOGICAL DEVICES AND ASSOCIATIONS, M-BUS MODELLING (GENERAL CONCEPT)

In DLMS/COSEM, metering equipment is modelled in physical and logical devices. The actual device is the physical device. The physical device can contain multiple logical devices. For this companion standard it is decided that there will be only 1 logical device (the management logical device).



2.1 Clients

The logical device has at least 3 associations: Public client (client Id 16), management client (client Id 1) and pre-established client (client Id 102).

There will be no direct connection to the M-bus device (used by P2) via the P3 interface. Access to the M-bus device has to be done with the objects offered by the logical device.

2.1.1 *Public client*

Public client is for test purpose. Due to the fact that the public client is with lowest security (no security), it must not allowed to read metering data, or perform any programming.

The services which must be supported within the Public client are:

- Block transfer with Get
- Get
- Selective access

2.1.2 *Pre established client*

The pre-established client is for broadcast purposes. It has to be used by the data concentrator, each time a broadcast service is needed.

The services which must be supported within the Pre established client are:

- Block transfer with Set
- Set
- Selective access
- Action

2.1.3 *Management client*

The management client (Client Id 1) is the client to be used by the data concentrator or the central access server in the case of GPRS and Ethernet meters, for regular point to point connections with the meters. This client must be able to perform any allowed operation on the devices, gathering with its own privileges, the ones of the public client and the pre-established client.

Additionally, this client is the addressee of the event notification request. It has Get, Set, Action and Event notification facilities.

The services which must be supported are within the Management client::

- Block transfer with Get
- Block transfer with Set
- Set
- Get
- Selective access
- Event notification
- Action

2.2 Access right

For each client, access right of each attribute for all the objects defined inside this document must be managed. This concerns Get, Set and Action access right.

According to the client, a given attribute may not be accessible either for Get or Set or Action facility. When the get, set or action service is not managed for an attribute, the result to return back in the case of an access attempt must be read-write-denied result. This must not be consider as a protocol error.

Management client must be able to manage all the necessary attributes access for a normal exploitation and maintenance of any network used.

There will be no direct connection to the M-bus device (used by P2) via the P3 interface. Access to the M-bus device has to be done with the objects offered by the logical device.

.

3 COMMUNICATION PROFILES AND SERVICES, SECURITY

This chapter gives the required and selected communication profiles and the security requirements to be implement for the P3 interface.

There are three communication profiles selected for the Dutch Smart meter.

These are:

- GPRS
- Ethernet
- PLC

DLMS/COSEM will use only the ‘pull’ mechanism for the application layer[3].

The description how the E-meter will establish a GPRS connection has been added to the profiles part (see chapter 3.1.1).

The profile for GPRS meters and Ethernet uses standard COSEM TCP/IP profile as defined in IEC 62056-47 or in DLMS UA Green Book edition 6. The TCP/UDP based profile is given in figure 3.1.

The DLMS/COSEM TCP based profile uses the standardized port number 4059.

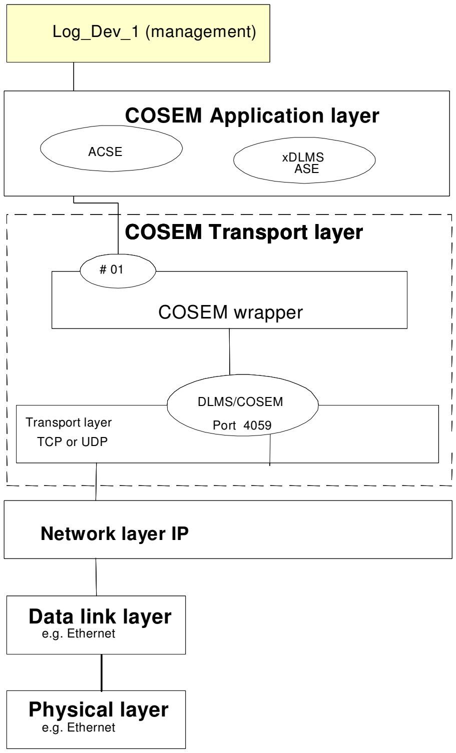


Figure 3.1: TCP/UDP based profile

[3] The operation principle used in this document is based on client / server model. Except for the Event notification which is based on unsolicited message sending for alarm management, for all the other services the client always asks for a request and the server send back the response of the concerned request. With respect to event notification, a GPRS meter shall be able to send an unsolicited message. The grid operator shall have the ability to configure which alarms will be sent unsolicited. Also channel set-up will be initiated by the GPRS meter.

The required setup objects for TCP/UDP, IPv4 and PPP will be described below. For the detailed attribute descriptions and methods see chapter 4.2 for the relevant paragraph of the Blue book [1].

TCP-UDP setup (Class ID: 41) **To set up the TCP or UDP sub-layer of the COSEM TCP or UDP based transport layer**			
1	Logical name	Octet-string	**0-b:25.0.0.255**
2	TCP-UDP_port	long-unsigned	Default 4059
3	IP_reference	octet-string	
4	MSS	long-unsigned	Min=40, max=65535, default=576
5	nb_of_sim_conn	unsigned	Value=1
6	inactivity_time_out	long-unsigned	Default=180

An instance of the TCP-UDP setup class contains all data necessary to set up the TCP or UDP sub-layer of the COSEM TCP or UDP based transport layer of a TCP-UDP/IP based communication profile.

In TCP-UDP/IP based communication profiles, all AAs between a physical device hosting one or more COSEM client application processes and a physical device hosting one or more COSEM server application processes rely on a single TCP or UDP connection. The TCP or UDP entity is wrapped in the COSEM TCP-UDP based transport layer. Within a physical device, each application process – client application process or server logical device - is bound to a Wrapper Port (WPort). The binding is done with the help of the SAP Assignment object.

On the other hand, a COSEM TCP or UDP based transport layer may be capable to support more than one TCP or UDP connections, between a physical device and several peer physical devices hosting COSEM application processes.

NOTE When a COSEM physical device supports various data link layers (for example Ethernet and PPP), then an instance of the TCP-UDP setup object is necessary for each of them.

IPv4 setup (Class ID: 42) **Handles all information that is related to the IP Address settings associated to a given device and to a lower layer connection on which these settings are used.**			
1	Logical name	octet-string	**0-b:25.1.0.255**
2	DL_reference port	octet-string	
3	IP_address	double-long-unsigned	
4	multicast_IP_address	array	
5	IP-options	array	
6	Subnet_mask	double-long-unsigned	
7	gateway_IP_address	double-long-unsigned	
8	use_DHCP_flag	boolean	
9	primary_DNS_address	double-long-unsigned	
10	secondary_DNS_address	double-long-unsigned	

An instance of the IPv4 setup class handles all information that is related to the IP Address settings associated to a given device and to a lower layer connection on which these settings are used.

There shall be an instance of this class in a device for each different network interfaces implemented.

PPP setup (Class ID: 44) **Handles all information that is related to PPP settings associated to a given physical device and to a lower layer connection on which these settings are used.**			
1	logical_name	Octet-string	**0-b:25.3.0.255**
2	PHY_reference	Octet string	
3	LCP_options	LCP_options_type	
4	IPCP_options	IPCP_options_type	
5	PPP_authentication[4]	PPP_auth_type	

There shall be an instance of this class for each network interface of a physical device, using the PPP protocol.

3.1 Communication profiles

The three specific communication profiles are given in the following paragraphs

3.1.1 GPRS communication profile

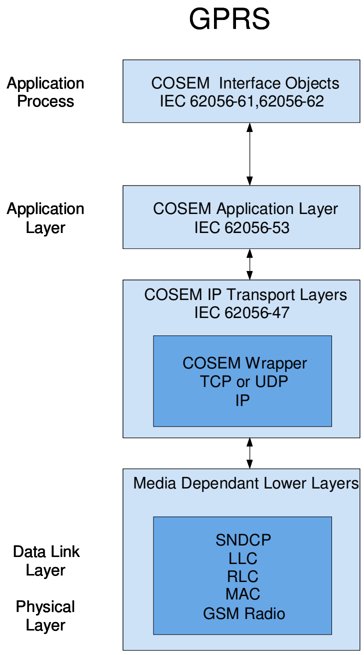

GPRS
Application Process
COSEM Interface Objects IEC 62056-61,62056-62
Application Layer
COSEM Application Layer IEC 62056-53
COSEM IP Transport Layers IEC 62056-47
COSEM Wrapper TCP or UDP IP
Media Dependant Lower Layers
Data Link Layer
Physical Layer
SNDCP LLC RLC MAC GSM Radio

Figure 3.2: GPRS communication profile

[4] Be aware that if care is not brought to these attribute, even they are managed by the manufacturers, they may be unusable. For example: as the same way the length of the meter Id is fixed, the specification must determine the maximum length of user name & password too

A “GPRS modem setup” object stores all the necessary data for a GPRS modem management. For the detailed attribute descriptions and methods see chapter 4.2 for the relevant paragraph of the Blue book [1].

GPRS modem setup (Class ID: 45) **A “GPRS modem setup” object stores all the necessary data for a GPRS modem management.**			
1	logical_name	octet-string	**0-b:25.4.0.255**
2	APN	octet-string	
3	PIN_code	long-unsigned	
4	quality_of_service	structure	

The connection setup mechanism used by the grid operators will be described in this chapter!!

3.1.2 Ethernet communication profile

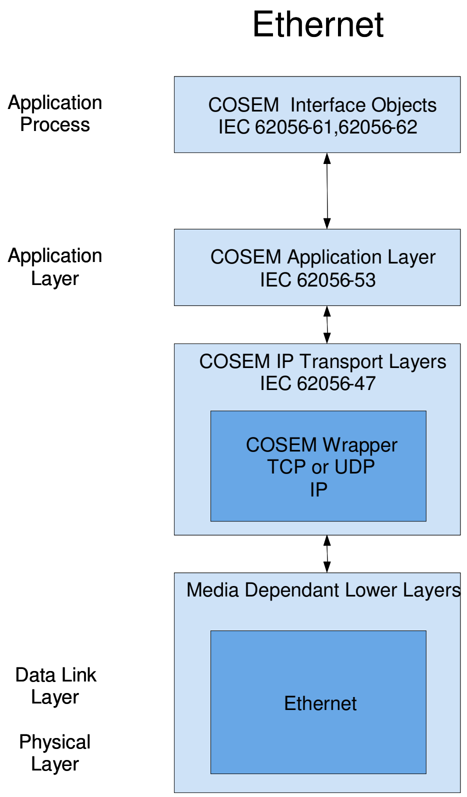

Ethernet
Application Process
COSEM Interface Objects IEC 62056-61,62056-62
Application Layer
COSEM Application Layer IEC 62056-53
COSEM IP Transport Layers IEC 62056-47
COSEM Wrapper TCP or UDP IP
Media Dependant Lower Layers
Data Link Layer
Physical Layer
Ethernet

Figure 3.3: Ethernet communication profile

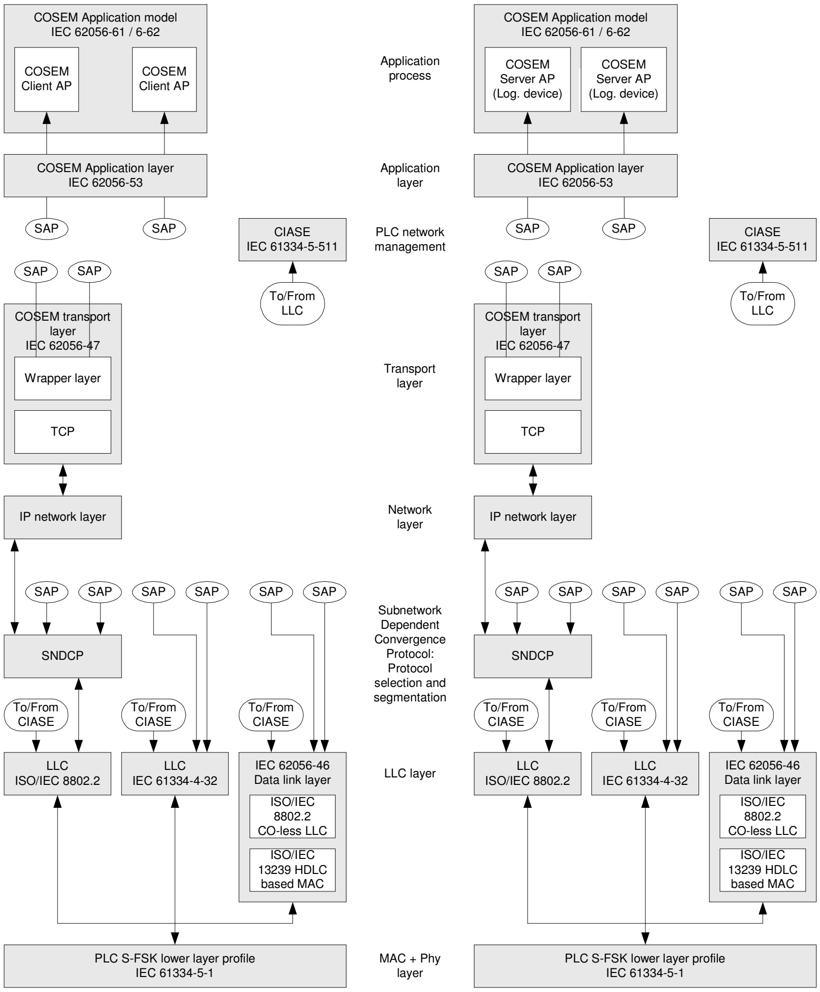
Ethernet setup (Class ID: 43) **Handles all information that is related to Ethernet settings associated to a given physical device and to a lower layer connection on which these settings are used**			
1	logical_name	Octet-string	**0-b:25.3.0.255**
2	MAC_address	Octet-string	No default value

An instance of the Ethernet setup class handles all information that is related to Ethernet settings associated to a given physical device and to a lower layer connection on which these settings are used.

There shall be an instance of this class for each network interface of a physical device, using the Ethernet protocol.

3.1.3 PLC

The profile for meters using the PLC medium as defined in IEC 62056-47 (has to be updated) or in DLMS UA Green Book edition 7 (in development) is pointed out below.


COSEM Application model IEC 62056-61 / 6-62
COSEM Client AP
COSEM Client AP
Application process
COSEM Application model IEC 62056-61 / 6-62
COSEM Server AP (Log. device)
COSEM Server AP (Log. device)
COSEM Application layer IEC 62056-53
Application layer
COSEM Application layer IEC 62056-53
SAP
CIASE IEC 61334-5-511
PLC network management
CIASE IEC 61334-5-511
To/From LLC
COSEM transport layer IEC 62056-47
Wrapper layer
TCP
Transport layer
IP network layer
Network layer
Subnetwork Dependent Convergence Protocol: Protocol selection and segmentation
SNDCP
To/From CIASE
LLC ISO/IEC 8802.2
LLC IEC 61334-4-32
IEC 62056-46 Data link layer
ISO/IEC 8802.2 CO-less LLC
ISO/IEC 13239 HDLC based MAC
LLC layer
PLC S-FSK lower layer profile IEC 61334-5-1
MAC + Phy layer

The ISO/IEC 8802.2 LLC layer may co-exist with IEC 61334-4-32 LLC layer and IEC 62056-46 data link layer used in existing implementations. On the server side, only one of them may be implemented. On the Client side, one or more of them may be implemented.The Client (concentrator) finds out the kind of LLC layer used by the server during the Discovery / Registration process and maps the LLC type to the System title.

The PLC solution requires support of ISO/IEC 8802.2, allowing to plug in TCP/IP later. This can co-exist with the IEC 61334-4-32 LLC layer and the IEC 62056-46 data link layer used in existing implementations.

The selected LLC layer is ISO/IEC 8802-2 LLC layer. The other two are only allowed for backwards compatibility reasons.

The nine PLC setup objects are given below with their OBIS codes. For the detailed attribute descriptions and methods see appendix B and [23].

S-FSK Phy&MAC setup (Class ID: 50)[5]			
1	Logical name	Octet-string	**0-0:26.0.0.255**

S-FSK Active Initiator (Class ID: 51)			
1	Logical name	Octet-string	**0-0:26.1.0.255**

S-FSK MAC synchronisation timeouts (Class ID: 52)			
1	Logical name	Octet-string	**0-0:26.2.0.255**

S-FSK MAC counters (Class ID: 53)			
1	Logical name	Octet-string	**0-0:26.3.0.255**

S-FSK IEC 61334-4-32 LLC Setup (Class ID: 55)			
1	Logical name	Octet-string	**0-0:26.5.0.255**

S-FSK Reporting system list (Class ID: 56)			
1	Logical name	Octet-string	**0-0:26.6.0.255**

ISO/IEC 8802-2 LLC Type 1 setup (Class ID: 57)			
1	Logical name	Octet-string	**0-0:27.0.0.255**

ISO/IEC 8802-2 LLC Type 2 setup (Class ID: 58)			
1	Logical name	Octet-string	**0-0:27.1.0.255**

ISO/IEC 8802-2 LLC Type 3 setup (Class ID: 59)			
1	Logical name	Octet-string	**0-0:27.2.0.255**

[5] The choice of the modulation band (attribute 6 ➔ frequency) (see appendix B.1) must be mutual agreed between the vendors to reach operability. A selection of which modulation band to use, shall be tuned after practical experiences in the field . The selection depends of the architecture of the Dutch Grid in general.

3.2 COSEM Security

Both access and message security are required to address the requirement for confidentiality and authentication in DLMS/COSEM communication protocol. COSEM Security should be implemented as described in DLMS UA Green Book edition 7 and DLMS UA Blue Book edition 9. Because these documents are not yet published the COSEM security requirements and way to implement are given in reference [22].

Access Security

DLMS/COSEM access security provides Low Level Security (LLS) and High Level Security (HLS) authentication mechanisms. LLS enable client only authentication with password and HLS enable client/server authentication with challenge mechanism. Authentication is used for association establishing. Associations offer specific view to COSEM objects with associated access rights to individual objects.

During the setup of the connection (the application association) **HLS** should be used. The Management Logical device should be accessible without security.

Message Security

DLMS/COSEM message security provides encryption and authentication of the COSEM APDU with standard symmetric key algorithms. It provides combined confidentiality and authentication using **GCM** cipher mode of operation.

Encryption is used to provide confidentiality for data. The data to be protected is called plaintext. Encryption transforms the data into Ciphertext. Ciphertext can be transformed back into plaintext using decryption.

Message authentication and encryption should be used in all situations and the **Advanced Encryption Standard (AES)** is the selected algorithm for ciphering.

Selected Security Suite:

Security Suite Id	Authentication Algorithm	Encryption Algorithm
7	**AES-GCM-128**	**AES-GCM-128**

Note: Security Suite 7 uses Combined Mode of operation with Authentication and Encryption. This choice covers extensive security at authentication and encryption level and realise full interoperability.

4 OVERVIEW OF OBJECT MODEL

The object model of the Dutch smart meter is divided in three parts:

a. Abstract objects (chapter 5)
b. Electricity related objects (chapter 6)
c. M-bus related objects (chapter 7)

Before all the required objects for the Dutch smart meter are described in more detail in the mentioned chapters an overview will be given of all required profiles. Further an explanation of the event and error handling is described in paragraph 4.2 of this chapter for better understanding how the required objects for covering that functionality are related to each other.

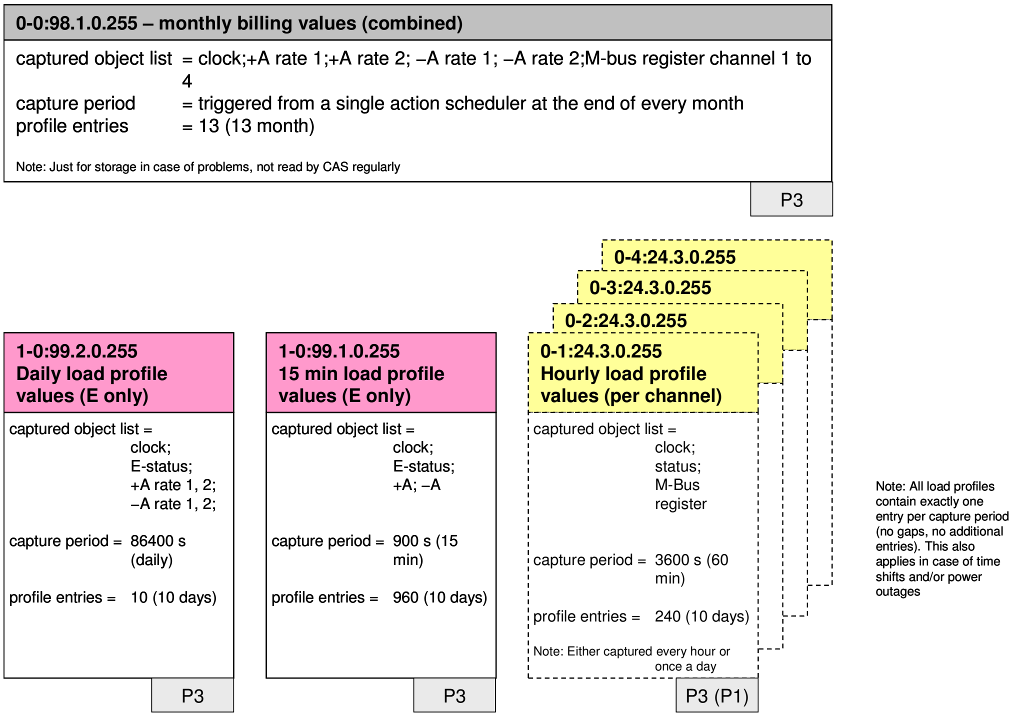
4.1 Profile structure Dutch smart meter

In figure 4.1 the profiles are summarized which will be required in relation to the Dutch smart meter requirements.

The "*Monthly billing values (combined)"* object is described in more detail in paragraph 5.5 (Abstract objects).

The "*Daily load profile values (E only)"* and "*15 min load profile values"* are described in more detail in paragraph 6.3 (Electricity related objects).

The "*Hourly load profile values (per channel)"* are part of paragraph 7.4 (M-bus related objects).



0-0:98.1.0.255 – monthly billing values (combined)

captured object list = clock;+A rate 1;+A rate 2; –A rate 1; –A rate 2;M-bus register channel 1 to 4
capture period = triggered from a single action scheduler at the end of every month
profile entries = 13 (13 month)

Note: Just for storage in case of problems, not read by CAS regularly

P3

0-4:24.3.0.255

0-3:24.3.0.255

0-2:24.3.0.255

1-0:99.2.0.255
Daily load profile values (E only)

captured object list = clock; E-status; +A rate 1, 2; –A rate 1, 2;

capture period = 86400 s (daily)

profile entries = 10 (10 days)

P3

1-0:99.1.0.255
15 min load profile values (E only)

captured object list = clock; E-status; +A; –A

capture period = 900 s (15 min)

profile entries = 960 (10 days)

P3

0-1:24.3.0.255
Hourly load profile values (per channel)

captured object list = clock; status; M-Bus register

capture period = 3600 s (60 min)

profile entries = 240 (10 days)

Note: Either captured every hour or once a day

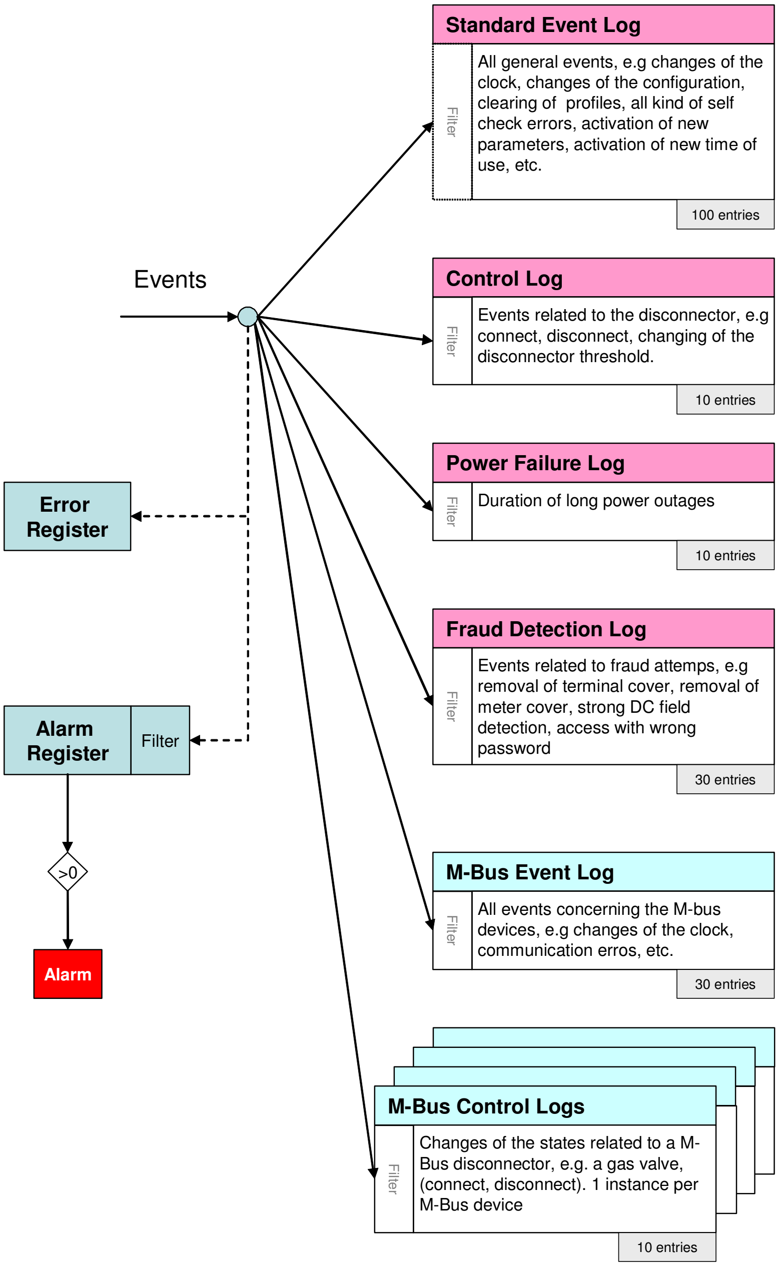
P3 (P1)

Note: All load profiles contain exactly one entry per capture period (no gaps, no additional entries). This also applies in case of time shifts and/or power outages

Figure 4.1: Structure of profiles of Dutch Smart meter

4.2 Event and error handling

This paragraph gives an overview of the event and error handling based on DLMS objects for the Dutch smart meter to fulfil the requirements stated in [21].


Standard Event Log
Filter
All general events, e.g changes of the clock, changes of the configuration, clearing of profiles, all kind of self check errors, activation of new parameters, activation of new time of use, etc.
100 entries
Events
Control Log
Filter
Events related to the disconnector, e.g connect, disconnect, changing of the disconnector threshold.
10 entries
Power Failure Log
Filter
Duration of long power outages
10 entries
Error Register
Fraud Detection Log
Filter
Events related to fraud attemps, e.g removal of terminal cover, removal of meter cover, strong DC field detection, access with wrong password
30 entries
Alarm Register
Filter
>0
Alarm
M-Bus Event Log
Filter
All events concerning the M-bus devices, e.g changes of the clock, communication erros, etc.
30 entries
M-Bus Control Logs
Filter
Changes of the states related to a M-Bus disconnector, e.g. a gas valve, (connect, disconnect). 1 instance per M-Bus device
10 entries

A lot of events are generated by the meter itself or by its environment. All these events are logged in several event logs. The objects are described in chapters 5, 6 and 7 in more detail. Additionally they are also used to set and clear errors as well as to trigger alarms. The diagram above gives an overview about the handling, the details of the objects are described in the chapters 5, 6 and 7.

4.2.1 Events

Every event has a unique code to identify the action which has triggered it. Every event is assigned to one event log (event filter) and it is only stored there. This assignment is fixed and can't be changed dynamically. Therefore the event filter is not visible from outside.

All event codes written in *italics* are optional (i.e. not required explicitly in NTA).

Event codes:

Number	**Name**	**Description**	**Standard Event log**	**Fraud Detection Log**	**Disconnector Control Log**	**M-Bus Event Log**	**M-Bus Control Log 1**	**M-Bus Control Log 2**	**M-Bus Control Log 3**	**M-Bus Control Log 4**
255	Event log cleared	Indicates that the event log was cleared. This is always the first entry in an event log. It is only stored in the affected event log.	x	x	x	x	x	x	x	x
1	*Power Down*	*Indicates a complete power down of the device. Please note that this is related to the device and not necessarily to the network.*	*x*							
2	*Power Up*	*Indicates that the device is powered again after a complete power down.*	*x*							
3	Daylight saving time enabled or disabled	Indicates the regular change from and to daylight saving time. The time stamp shows the time before the change. This event is not set in case of manual clock changes and in case of power failures.	x							
4	Clock adjusted (old date/time)	Indicates that the clock has been adjusted. The date/time that is stored in the event log is the old date/time before adjusting the clock.	x							
5	Clock adjusted (new date/time)	Indicates that the clock has been adjusted. The date/time that is stored in the event log is the new date/time after adjusting the clock.	x							
6	Clock invalid	Indicates that clock may be invalid, i.e. if the power reserve of the clock has exhausted. It is set at power up.	x							
7	Replace Battery	Indicates that the battery must be exchanged due to the expected end of life time.	x							
8	*Battery voltage low*	*Indicates that the current battery voltage is low.*	*x*							
9	TOU activated	Indicates that the passive TOU has been activated.	x							
10	Error register cleared	Indicates that the error register was cleared.	x							
11	Alarm register cleared	Indicates that the alarm register was cleared.	x							
12	Program memory error	Indicates a physical or a logical error in the program memory.	x							
13	RAM error	Indicates a physical or a logical error in the RAM	x							
14	NV memory error	Indicates a physical or a logical error in the non volatile memory	x							
15	Watchdog error	Indicates a watch dog reset or a hardware reset of the microcontroller.	x							
16	Measurement system error	Indicates a logical or physical error in the measurement system	x							

Number	Name	Description	Standard Event log	Fraud Detection Log	Disconnector Control Log	M-Bus Event Log	M-Bus Control Log 1	M-Bus Control Log 2	M-Bus Control Log 3	M-Bus Control Log 4
17	Firmware ready for activation	Indicates that the new firmware has been successfully downloaded and verified, i.e. it is ready for activation	X							
18	Firmware activated	Indicates that a new firmware has been activated	X							
19-39	reserved for future use									
40	Terminal cover removed	Indicates that the terminal cover has been removed		X						
41	Terminal cover closed	Indicates that the terminal cover has been closed		X						
42	Strong DC field detected	Indicates that a strong magnetic DC field has been detected.		X						
43	No strong DC field anymore	Indicates that the strong magnetic DC field has disappeared.		X						
44	Meter cover removed	Indicates that the meter cover has been removed.		X						
45	Meter cover closed	Indicates that the meter cover has been closed.		X						
46	n times wrong password	Indicates that a user tried to gain access with a wrong password (intrusion detection)		X						
47-59	reserved for future use									
60	Manual disconnection	Indicates that the disconnector has been manually disconnected.			X					
61	Manual connection	Indicates that the disconnector has been manually connected.			X					
62	Remote disconnection	Indicates that the disconnector has been remotely disconnected.			X					
63	Remote connection	Indicates that the disconnector has been remotely connected.			X					
64	Local disconnection	Indicates that the disconnector has been locally disconnected (i.e. via the limiter).			X					
65	Limiter threshold exceeded	Indicates that the limiter threshold has been exceeded.			X					
66	Limiter threshold ok	Indicates that the monitored value of the limiter dropped below the threshold.			X					
67	Limiter threshold changed	Indicates that the limiter threshold has been changed			X					
68-100	reserved for future use									
100	Communication error M-Bus channel 1	Indicates a communication problem when reading the meter connected to channel 1 of the M-Bus				X				
101	Communication ok M-Bus channel 1	Indicates that the communication with the M-Bus meter connected to channel 1 of the M-Bus is ok again.				X				
102	Replace Battery M-Bus channel 1	Indicates that the battery must be exchanged due to the expected end of life time.				X				
103	Fraud attempt M-Bus channel 1	Indicates that a fraud attempt has been registered.				X				

Number	Name	Description	Standard Event log	Fraud Detection Log	Disconnector Control Log	M-Bus Event Log	M-Bus Control Log 1	M-Bus Control Log 2	M-Bus Control Log 3	M-Bus Control Log 4
104	Clock adjusted M-Bus channel 1	Indicates that the clock has been adjusted.				X				
105-109	reserved for future use									
110	Communication error M-bus channel 2	Indicates a communication problem when reading the meter connected to channel 2 of the M-Bus				X				
111	Communication ok M-bus channel 2	Indicates that the communication with the M-Bus meter connected to channel 2 of the M-Bus is ok again.				X				
112	Replace Battery M-Bus channel 2	Indicates that the battery must be exchanged due to the expected end of life time.				X				
113	Fraud attempt M-Bus channel 2	Indicates that a fraud attempt has been registered in the M-Bus device.				X				
114	Clock adjusted M-Bus channel 2	Indicates that the clock has been adjusted.				X				
115-119	reserved for future use									
120	Communication error M-bus channel 3	Indicates a communication problem when reading the meter connected to channel 3 of the M-Bus				X				
121	Communication ok M-bus channel 3	Indicates that the communication with the M-Bus meter connected to channel 3 of the M-Bus is ok again.				X				
122	Replace Battery M-Bus channel 3	Indicates that the battery must be exchanged due to the expected end of life time.				X				
123	Fraud attempt M-Bus channel 3	Indicates that a fraud attempt has been registered.				X				
124	Clock adjusted M-Bus channel 3	Indicates that the clock has been adjusted.				X				
125-129	reserved for future use									
130	Communication error M-bus channel 4	Indicates a communication problem when reading the meter connected to channel 4 of the M-Bus				X				
131	Communication ok M-bus channel 4	Indicates that the communication with the M-Bus meter connected to channel 4 of the M-Bus is ok again.				X				
132	Replace Battery M-Bus channel 4	Indicates that the battery must be exchanged due to the expected end of life time.				X				
133	Fraud attempt M-Bus channel 4	Indicates that a fraud attempt has been registered.				X				
134	Clock adjusted M-Bus channel 4	Indicates that the clock has been adjusted.				X				
144-159	reserved for future use									
160	Manual disconnection M-Bus channel 1	Indicates that the disconnector has been manually disconnected.					X			

Number	Name	Description	Standard Event log	Fraud Detection Log	Disconnector Control Log	M-Bus Event Log	M-Bus Control Log 1	M-Bus Control Log 2	M-Bus Control Log 3	M-Bus Control Log 4
161	Manual connection M-Bus channel 1	Indicates that the disconnector has been manually connected.					X			
162	Remote disconnection M-Bus channel 1	Indicates that the disconnector has been remotely disconnected.					X			
163	Remote disconnection M-Bus channel 1	Indicates that the disconnector has been remotely connected.					X			
164	Valve alarm M-Bus channel 1	Indicates that a valve alarm has been registered.					X			
165-169	reserved for future use									
170	Manual disconnection M-Bus channel 2	Indicates that the disconnector has been manually disconnected.						X		
171	Manual connection M-Bus channel 2	Indicates that the disconnector has been manually connected.						X		
172	Remote disconnection M-Bus channel 2	Indicates that the disconnector has been remotely disconnected.						X		
173	Remote disconnection M-Bus channel 2	Indicates that the disconnector has been remotely connected.						X		
174	Valve alarm M-Bus channel 2	Indicates that a valve alarm has been registered.						X		
175-179	reserved for future use									
180	Manual disconnection M-Bus channel 3	Indicates that the disconnector has been manually disconnected.							X	
181	Manual connection M-Bus channel 3	Indicates that the disconnector has been manually connected.							X	
182	Remote disconnection M-Bus channel 3	Indicates that the disconnector has been remotely disconnected.							X	
183	Remote disconnection M-Bus channel 3	Indicates that the disconnector has been remotely connected.							X	
184	Valve alarm M-Bus channel 3	Indicates that a valve alarm has been registered.							X	
185-189	reserved for future use									
190	Manual disconnection M-Bus channel 4	Indicates that the disconnector has been manually disconnected.								X
191	Manual connection M-Bus channel 4	Indicates that the disconnector has been manually connected.								X
192	Remote disconnection M-Bus channel 4	Indicates that the disconnector has been remotely disconnected.								X
193	Remote disconnection M-Bus channel 4	Indicates that the disconnector has been remotely connected.								X

Number	**Name**	**Description**	**Standard Event log**	**Fraud Detection Log**	**Disconnector Control Log**	**M-Bus Event Log**	**M-Bus Control Log 1**	**M-Bus Control Log 2**	**M-Bus Control Log 3**	**M-Bus Control Log 4**
194	Valve alarm M-Bus channel 4	Indicates that a valve alarm has been registered.								x
195-229	reserved for future use									
230-249	manufacturer specific	Manufacturer specific events can be registered								
250-254	reserved for future use									

The event code is not available as a register. To identify the event code in the different event logs, the following OBIS codes are used in the captured_object_list of the event log:

0-0:96.11.0.255	Standard event Log
0-0:96.11.1.255	Fraud detection Log
0-0:96.11.2.255	Disconnector control Log
0-0:96.11.3.255	M-Bus event log
0-1:96.11.4.255	M-Bus control log 1
0-2:96.11.4.255	M-Bus control log 2
0-3:96.11.4.255	M-Bus control log 3
0-4:96.11.4.255	M-Bus control log 4

4.2.2 Event Logs

The E-meter features 4 different event logs as described below. Additionally there is one event log for all M-Bus devices as well as one control log per M-Bus channel available.
All logs except the power failure log have the same basic structure (timestamp and event code). The structure per event log is fixed, i.e. it is not possible to store different parameters per event.

Standard Event Log [0-0:99.98.0.255] (paragraph 5.7)

Contains all events not recorded in a special event log, e.g. changes of the clock, changes of the configuration, clearing of profiles, all kind of self check errors, activation of new parameters, activation of new time of use, etc.
Structure: Timestamp – Event Code
Minimum size: 100 entries

Control Log [0-0:99.98.2.255] (paragraph 5.10)

Contains all events related to the disconnector, e.g. connect, disconnect, changing of the disconnector threshold.
Structure: Timestamp – Event Code – Currently active disconnector threshold
Minimum size: 10 entries

Power Failure Log [1-0:99.97.0.255]) (paragraph 5.9)

Contains all events related to long power outages, i.e. start and end of a long power outage.
Structure: Timestamp – Event Code
Minimum size: 10 entries

Fraud Detection Log [0-0:99.98.1.255] (paragraph 5.8)

Contains all events related to the detection of fraud attempts, e.g. removal of terminal cover, removal of meter cover, strong DC field detection, access with wrong password, etc.
Structure: Timestamp – Event Code
Minimum size: 30 entries

M-Bus Event Log [0-0:99.98.3.255]   (paragraph 7.5)

Contains all events related to the M-Bus devices, e.g. changes of the clock, communication errors, etc.
Structure: Timestamp – Event Code
Minimum size: 30 entries

M-Bus Control Logs (0-x:24.5.0.255) (paragraph 7.6)

Contains all events related to an M-Bus disconnector, e.g. a gas valve (open valve, close valve).
Structure: Timestamp – Event Code
Minimum size: 10 entries

4.2.3 Error Handling

A predefined selection of events set and clear flags in the error register. The error register can be read and displayed at anytime to see, if there is a malfunction in the device.

Depending on the type of error, some errors clear themselves if the reason for the error has disappeared. Other must be cleared via CAS. Nevertheless the events are stored in one of the event logs.

Error Codes

The table below gives an overview of all errors and their assignment.

reserved								Com Errors								Critical Errors								Other Errors							
Byte 4								Byte 3								Byte 2								Byte 1							
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
not used	not used	not used	not used	not used	not used	not used	not used	not used	not used	not used	not used	Communication error M-Bus channel 4	Communication error M-Bus channel 3	Communication error M-Bus channel 2	Communication error M-Bus channel 1	not used	not used	not used	Watchdog error	Measurement System Error	NV memory Error	RAM Error	Program memory error	not used	not used	not used	not used	not used	not used	Replace battery	Clock invalid

For a detailed description see the corresponding event. Critical errors and the replacement of the battery must be cleared via CAS, all other clear themselves if the corresponding error condition has disappeared.

4.2.4 Alarm Handling

A selection of events can be made which are treated as alarms (alarm filter). If one of these events occurs, the corresponding flag in the alarm register is set and an alarm is then raised via PLC or via GSM/GPRS.

All alarm flags in the alarm register (0-0:97.98.0.255) remain active until the alarm register is cleared via CAS (acknowledgment).

Typically fraud attempts are selected as alarm triggers.

Typically fraud attempts and critical errors are selected as alarm triggers. Power outages normally can't be selected since the communication network is also down in case of a power outage.

Alarm Codes

The table below gives an overview of all possible alarms and their assignment.

reserved								M-Bus Errors								Critical Errors								Other Alarms							
Byte 4								Byte 3								Byte 2								Byte 1							
8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1	8	4	2	1
not used	not used	not used	not used	not used	not used	not used	not used	Fraud attempt M-Bus channel 4	Fraud attempt M-Bus channel 3	Fraud attempt M-Bus channel 2	Fraud attempt M-Bus channel 1	Communication error M-Bus channel 4	Communication error M-Bus channel 3	Communication error M-Bus channel 2	Communication error M-Bus channel 1	not used	not used	Fraud attempt	Watchdog error	Measurement system error	NV memory error	RAM error	Program memory error	not used	not used	not used	not used	not used	not used	Replace battery	Clock invalid

All fraud attempts are grouped, i.e. for alarming it is not necessary to see the exact type of fraud which caused the alarm. This can be found out by checking the error register or the appropriate event log.

Alarm Filters

Depending on the capabilities of the CAS and the policy of the utility, not all possible alarms are wanted. Therefore an alarm filter (0-0:97.98.10.255) can be programmed to mask out unwanted alarms.

The structure of the filter is the same as for the alarm codes.

4.2.5 AMR Profile status Code

In all load profiles a new simplified status code is used for every entry. It is derived from the existing VDN status word but can only be used for profiles containing cumulative values.

The AMR Profile status code has a size of 1 byte and it is shown in hexadecimal form.

The following table describes the state and the function of all bits:

lag	Description
Bit 7 PDN	**Power down**: This bit is set to indicate that an all-phase power failure occurred.
Bit 6	not used
Bit 5 CAD	**Clock adjusted**: The bit is set when clock has been adjusted more than the synchronisation limit. A the same time the DNV flag is set because the capture period deviates from its nominal lengths and may not be used for billing.
Bit 4	not used.
Bit 3 DST	**Daylight saving**: Indicates whether or not the daylight saving time is currently active. The bit is set if the daylight saving time is active (summer) and cleared in winter.
Bit 2 DNV	**Data not valid**: Indicates that the current entry may not be used for billing e.g. due to time shift or if no values have been recorded during the capture period.
Bit 1 CIV	**Clock invalid**: The power reserve of the calendar clock has been exhausted. The time is declared as invalid. A the same time the DNV bit is set.
Bit 0 ERR	**Critical error**: A serious error such as a hardware failure or a checksum error has occurred. A the same time, the DNV bit is set.

5 ABSTRACT OBJECTS

5.1 SAP assignment, Association LN,COSEM logical device name

SAP Assignment (Class ID: 17)			
Information about the logical devices in the physical device			
1	Logical name	Octet-string	**0-0:41.0.0.255**
2	Value	asslist_type	Only 1 logical device: The management logical device

Association LN (Class ID: 15)			
1	logical_name	octet-string	**0-0:40.0.0.255**
2	object_list	objlist_type	
3	associated_partners_id	associated_partners_type	
4	application_context_ name	application_context_name	
5	xDLMS_context_info	xDLMS_context_type	
6	authentication_ mechanism_name	mechanism_name	
7	LLC_secret	octet-string	
8	association_status	enum	
1	reply_to_HLS_ authentication		

COSEM Logical Device Name (Class ID: 1)			
1	Logical name	Octet-string	**0-0:42.0.0.255**
2	Value	Octet-string	Unique identification of the logical device

5.2 Identification numbers

Device ID 1 (Class ID: 1)			
1	Logical name	Octet-string	**0-0:96.1.0.255**
2	Value	Octet-string[16]	E-meter serial number (Serial number of the device, handled by the manufacturer)

Device ID 2 (Class ID: 1)			
1	Logical name	Octet-string	**0-0:96.1.1.255**
2	Value	Octet-string[48]	E-meter equipment identifier (Owned and handled by the utility, has no meaning to the device.)

Device ID 3 (Class ID: 1)			
1	Logical name	Octet-string	**0-0:96.1.2.255**
2	Value	Octet-string[48]	Function location (Owned and handled by the utility, has no meaning to the device.)

Device ID 4 (Class ID: 1)			
1	Logical name	Octet-string	**0-0:96.1.3.255**
2	Value	Octet-string[48]	Location information (Owned and handled by the utility, has no meaning to the device.)

Device ID 5 (Class ID: 1)			
1	Logical name	Octet-string	**0-0:96.1.4.255**
2	Value	Octet-string[48]	No special meaning defined (General purpose ID for any identification purposes. Owned and handled by the utility, has no meaning to the device.)

5.3 Clock

Clock (Class ID: 8)			
1	Logical name	Octet-string	**0-0:1.0.0.255**
2	time	octet-string	current local date and time
3	time_zone	long	
4	status	status	
5	daylights_savings_begin	octet-string	last Sunday in March at 02:00
6	daylights_savings_end	octet-string	last Sunday in October at 03:00
7	daylights_savings_deviation	integer	Value = 60
8	daylights_savings_enabled	boolean	
9	clock_base	enum	Value = 1. Internal crystal

Clock Time Shift Limit (Class ID: 3)			
1	Logical name	Octet-string	**1-0:0.9.11.255**
2	value	unsigned	Value = 60. Maximum allowed time shift without registration of a time shift event
3	Scaler_unit	scal_unit_type	Value = {0,7}. Scaler=0, unit=seconds

5.4 Activity Calendar and Special Days Table

Activity Calendar (Class ID: 20) **Time of use for tariff control**			
1	Logical name	Octet-string	**0-0:13.0.0.255**
2	calendar_name_active	octet-string	
3	season_profile_active	array[4]	minimum of 4 seasons
4	week_profile_table_active	array[4]	minimum of 4 week profiles (= 1 week profile per season)
5	day_profile_table_active	array[4]	minimum of 4 day profiles (= weekday, Saturday, Sunday, special day). Every day profile can contain at least 4 entries (switching points)
6	calendar_name_passive	octet-string	
7	season_profile_passive	array[4]	see above
8	week_profile_table_passive	array[4]	see above
9	day_profile_table_passive	array[4]	see above
10	activate_passive_calendar_time	octet-string	immediate activation can be done by setting the activation date to the current date

Special Days Table (Class ID: 11)			
1	Logical name	Octet-string	**0-0:11.0.0.255**
2	Entries	array[30]	At least 30 special days

Currently active tariff (Class ID: 1)			
1	Logical name	Octet-string	**0-0:96.14.0.255**
2	Value	Octet-string	currently active tariff = active_mask of register activation object

5.5 Billing periods and profiles

End of billing period 1 (Class ID: 22) **End of every month**			
1	Logical name	Octet-string	**0-0:15.0.0.255**
2	executed_script	script	billing period reset
3	type	enum	Value = 1, fixed time, wildcard in date
4	execution_time	array	Value = "00000000"; "FFFFFF01FF", at midnight (or any other time) of the first day of every month

Data of billing period 1 (Class ID: 7) **Monthly billing values**			
1	Logical name	Octet-string	**0-0:98.1.0.255**
2	buffer	array	
3	capture_objects	array	{8,0-0:1.0.0.255,2,0}; {3,1-0:1.8.1.255,2,0}; {3,1-0:1.8.2.255,2,0}; {3,1-0:2.8.1.255,2,0}; {3,1-0:2.8.2.255,2,0}; {4,0-x:24.2.y.255,2,0}; ... (= clock;+A rate 1;+A rate 2;–A rate 1;–A rate 2; up to 4 M-Bus register values) Can be extended with up to 4 M-Bus register values or additional tariff registers
4	capture_period	double-long-unsigned	0, triggered from single action scheduler with billing period 1
5	sort_method	enum	1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥13 months

5.6 Error and Alarm Handling (error register, alarm register)

Error Object (Class ID: 1) **Error register**			
1	Logical name	Octet-string	**0-0:97.97.0.255**
2	Value	double-long-unsigned	Error code (See definition of error codes in paragraph 4.2.3)

Alarm Object (Class ID: 1) **Alarm register**			
1	Logical name	Octet-string	**0-0:97.98.0.255**
2	Value	double-long-unsigned	Alarm code (See definition of alarm codes in paragraph 4.2.4)

Alarm filter (Class ID: 1)			
1	Logical name	Octet-string	**0-0:97.98.10.255**
2	Value	double-long-unsigned	This filter defines, if an event is handled as an alarm when it appears. Bit mask with the same structure as the current value of the alarm object code (See definition of alarm filters in paragraph 4.2.4)

5.7 Event Handling (events and logs)

Event Log (Class ID: 7) **Standard event log containing errors and alarms**			
1	Logical name	Octet-string	**0-0:99.98.0.255**
2	buffer	array	
3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; {0-0:96.11.0.255,2,0} (= clock;event code) (See definition of event codes in paragraph 4.2.1)
4	capture_period	double-long-unsigned	0, asynchronously
5	sort_method	enum	1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥100

5.8 Fraud detection (event log)

Fraud Detection Log (Class ID: 7) **Event log containing all fraud detection events**			
1	Logical name	Octet-string	**0-0:99.98.1.255**
2	buffer	array	
3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; {1,0-0:96.11.1.255,2,0} (= clock;tamper event code) (See definition of event codes in paragraph 4.2.1)
4	capture_period	double-long-unsigned	0, asynchronously
5	sort_method	enum	1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥30

5.9 Power Failure (counter, thresholds and event log)

Number of power failures in any phases (Class ID: 1) **(single and polyphase meters)**			
1	Logical name	Octet-string	**0-0:96.7.21.255**
2	Value	long-unsigned	

Number of long power failures in any phases (Class ID: 1) **(single and polyphase meters)**			
1	Logical name	Octet-string	**0-0:96.7.9.255**
2	Value	long-unsigned	

Time threshold for long power failure (Class ID: 3)			
1	Logical name	Octet-string	**0-0:96.7.20.255**
2	Value	long-unsigned	Value = 180
3	scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds

Duration of long power failures in any phase (Class ID: 3)			
1	Logical name	Octet-string	**0-0:96.7.19.255**
2	Value	long-unsigned	Is reset at the end of the power failure, i.e. after capturing in the event log
3	scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds

Power Failure Event Log (Class ID: 7)			
1	Logical name	Octet-string	**1-0:99.97.0.255**
2	buffer	array	
3	capture_objects	Array [2]	{8,0-0:1.0.0,2,0}; {1,0-0:96.7.19.255,2,0} (= clock; duration of long power failures in any phase) Timestamp = end of power failure
4	capture_period	double-long-unsigned	0, asynchronously
5	sort_method	enum	1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥10

5.10 Disconnector (Control and Log, scheduler)

Disconnect control (Class ID: 70) Controls the connection and disconnection of the premises of the consumer			
1	Logical name	Octet-string	**0-0:96.3.10.255**
2	output_state	boolean	
3	control_state	enum	
4	control_mode	enum	
1	remote_disconnect		
2	remote_connect		

The Disconnect Control class is a new class that is not specified in the current version of the Blue Book [1], but it will be defined in the next version. The specification of this class can be found in Appendix A.3 and in [23].

Control log (Class ID:7) **Changes of the states related to the disconnect control are recorded (changing threshold, connect, disconnect)**			
1	logical_name	octet-string	**0-0:99.98.2.255**
2	buffer	array	
3	capture_objects	array	{8, 0-0:1.0.0, 2, 0}, clock; {1, 0-0: 96.11.2, 255,2, 0}, control event code {71, 1-0:17.0.0, 3, 0}, limiter threshold Event codes must be defined in chapter 4.2.1
4	capture_period	double-long-unsigned	Value = 0, asynchronously
5	sort_method	enum	Value = 1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥10
1	reset		
2	capture		optional, has no meaning

Disconnect Control Scheduler (Class ID: 22)			
1	Logical name	Octet-string	**0-0:15.0.1.255**
2	executed_script	script	connect or disconnect script of disconnector script table
3	type	enum	Value = 1, fixed time
4	execution_time	array	Time; date. Dedicated time point for connection or disconnection. No wildcards in date allowed

Disconnector Script Table (Class ID: 9)			
1	Logical name	Octet-string	**0-0:10.0.106.255**
2	Scripts	Array[2]	connect and disconnect service of the disconnector object

5.11 Limiter

Limiter (Class ID: 71) **Handles the normal current monitoring as well as the emergency settings (code red)**			
1	Logical name	Octet-string	**0-0:17.0.0.255**
2	monitored_value	value_definition	instantaneous current
3	threshold_active	threshold	
4	threshold_normal	threshold	
5	treshold_emergency	threshold	
6	min_over_threshold_duration	double_long_unsigned	
7	min_under_threshold_duration	double_long_unsigned	
8	emergency_profile	emergency_profile_type	
9	emergency_profile_group_id	long-unsigned	
10	emergency_profile_active	boolean	
11	actions	action_set	
1	emergency_profile_update		

The Limiter class is a new class that is not specified in the current version of the Blue Book [1], but it will be defined in the next version. The specification of this class can be found in Appendix A.2 and in [23].

5.12 P1 objects (messages, readout list)

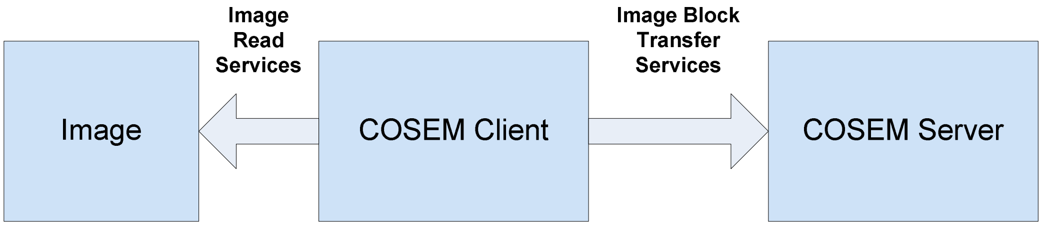
Consumer Message Text (Class ID: 1) **Consumer message text sent to port P1**			
1	Logical name	Octet-string	**0-0:96.13.0.255**
2	Value	Octet-string[1024]	Message text sent to port P1 without any further interpretation

Consumer Message Code (Class ID: 1) **Consumer message code shown on display and P1**			
1	Logical name	Octet-string	**0-0:96.13.1.255**
2	Value	Octet-string[64]	Message code must be shown on numeric display (including scroll feature), message codes are defined by GO.

General local port readout (Class ID: 7) **P1 port readout list**			
1	Logical name	Octet-string	**0-0:21.0.0.255**
2	buffer	array	last readout
3	capture_objects	Array [16]	readout objects, a maximum of 16 entries is possible
4	capture_period	double-long-unsigned	10, update period of P1 [s]
5	sort_method	enum	1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	1

5.13 Firmware upgrade

Image Block Transfer Mechanism is used to transfer Firmware Image(s) to electricity meters. Image Transfer Process from COSEM Client to COSEM Server uses Image Read Services to read ImageBlocks from Image and Image Block Transfer Services to transfer the ImageBlocks to COSEM Server.



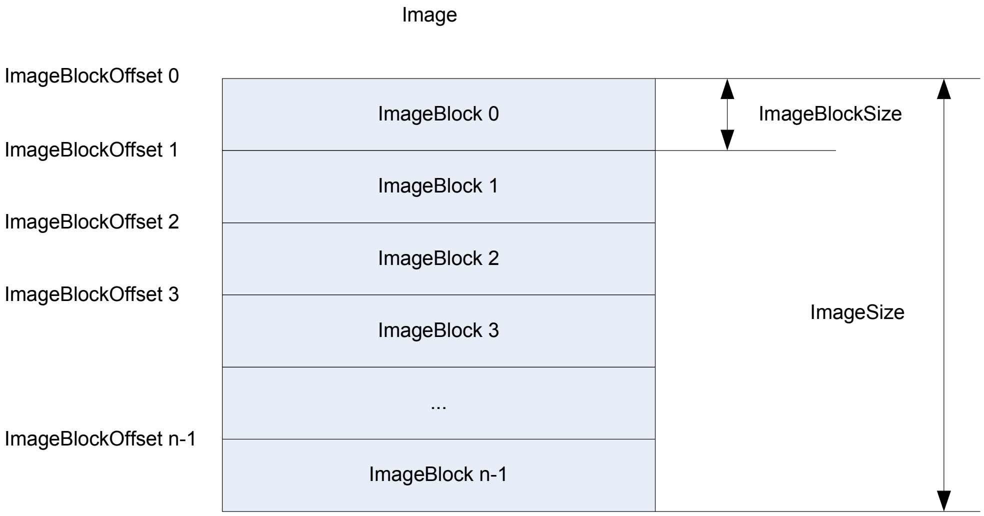


Image is data of specified size.

ImageSize is size of Image expressed in bytes. Image is divided into consecutive ImageBlocks of data of specified ImageBlockSize.

ImageBlock is part of Image with sequential ImageBlockNumber at specified ImageBlockOffset from the beginning of the Image.

ImageBlockSize is size of ImageBlock expressed in bytes.

ImageBlockNumber is sequential number of the ImageBlock starting from 0.

ImageBlockOffset is offset of the ImageBlock expressed in bytes from the beginning of the Image.

Image Transfer (Class ID: 18) **Allows transfer of Firmware Image(s) to COSEM servers**			
1	Logical_name	octet-string	**0-0:44.0.0.255**
2	Image_block_size	octet-string	
3	Image_block_transfer	image_block_transfer_type	
4	Image_missing_blocks	bit-string	
5	Image_first_missing_block_offset	double-long-unsigned	
6	Transfer_enabled	boolean	
7	Images_info	array	
1	Init_transfer		
2	Verify_image		
3	Verify_and _activate_image		

Reading ImageBlockSize

ImageBlockSize is the transfer unit that can be transferred to COSEM Server. Different COSEM Servers can support different ImageBlockSize. COSEM Client has to read ImageBlockSize parameter from COSEM Server to be able to transfer ImageBlocks with required ImageBlockSize.

Initiate Image Transfer

COSEM Client initiates Image Transfer on COSEM Server. Initiation is performed individually with each COSEM Server. After initiation COSEM Server is prepared to accept ImageBlocks.

Image Block Transfer

COSEM Client transfers all ImageBlocks in the Image to COSEM Server. Each ImageBlock is read from Image with Image Read Service and delivered to COSEM Server. ImageBlocks are delivered individually or with broadcast service where COSEM Servers with initiated Image Transfer accept ImageBlocks. Other COSEM Servers silently discard the ImageBlocks.

Filling Missing Blocks

COSEM Client fills ImageBlocks that have been missed from COSEM Servers with reading of missing ImageBlocks from COSEM Server and transferring the missing ImageBlocks again. Repeated transfer is performed for all missing ImageBlocks reported from COSEM Server. Filling Missing Blocks is performed individually with each COSEM Server. Missing ImageBlocks can be obtained by obtaining all missing ImageBlocks or obtaining first missing ImageBlock information.

Activate Image

COSEM Client activates Image to make it operational in COSEM Server.

The Image Transfer class is a new class that is not specified in the current version of the Blue Book [1], but it will be defined in the next version. The specification of this class can be found in Appendix C.1

Image Activation Scheduler (Class ID: 22) **Activate new firmware**			
1	Logical_name	octet-string	**0-0:15.0.2.255**
2	Executed_script	script	
3	Type	enum	1; fixed time, wildcard in date
4	Execution_time	array	Time; date; dedicated time point for activation

6 ELECTRICITY RELATED OBJECTS

This section holds a complete overview of all electricity related objects and their attributes.

6.1 Identification numbers (i.e. firmware version)

Active firmware version (Class ID: 1)			
1	Logical name	Octet-string	**1-0:0.2.0.255**
2	Value	Octet-string	Version identifier of the currently active firmware in the E-meter

6.2 E registers (+A, -A, all rate registers)

Active energy import (+A) (Class ID: 3)			
1	Logical name	Octet-string	**1-0:1.8.0.255**
2	Value	double-long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy export –A (Class ID: 3)			
1	Logical name	Octet-string	**1-0:2.8.0.255**
2	Value	double-long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy import (+A) rate 1 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:1.8.1.255**
2	Value	double-long-unsigned	Rate 1 = Low tariff
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy import (+A) rate 2 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:1.8.2.255**
2	Value	double-long-unsigned	Rate 2 = normal/high tariff
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy import (+A) rate 3 (Class ID: 3) **(for future use)**			
1	Logical name	Octet-string	**1-0:1.8.3.255**
2	Value	double-long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy import (+A) rate 4 (Class ID: 3) **(for future use)**			
1	Logical name	Octet-string	**1-0:1.8.4.255**
2	Value	double-long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy export (-A) rate 1 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:2.8.1.255**
2	Value	double-long-unsigned	Rate 1 = Low Tariff
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy export (-A) rate 2 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:2.8.2.255**
2	Value	double-long-unsigned	Rate 2 = Normal/high tariff
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy export (-A) rate 3 (Class ID: 3) **(for future use)**			
1	Logical name	Octet-string	**1-0:2.8.3.255**
2	Value	double-long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

Active energy export (-A) rate 4 (Class ID: 3) **(for future use)**			
1	Logical name	Octet-string	**1-0:2.8.4.255**
2	Value	double-long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh, resolution: 0.000 kWh

6.3 E profiles (15', daily)

Load profile with period 1 (Class ID: 7) **E interval readings every 15 minutes**			
1	Logical name	Octet-string	**1-0:99.1.0.255**
2	buffer	array	The buffer must be filled monotonously, i.e. no irregular entries are allowed = exactly one entry per capture period

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; {1,0-0:96.10.1.255,2,0}
			{3,0-0:1.8.0.255,2,0}; {3,0-0:2.8.0.255,2,0}
			(= clock; AMR profile status; +A;–A)
			
			Profile status ➔ see paragraph 4.2.5

4	capture_period	double-long-unsigned	900 (15 minutes)
5	sort_method	enum	1 or 3 (unsorted (FIFO) or sorted (largest))
6	sort_object	object definition	none or {8,0-0:1.0.0.255,2,0} (unsorted or sorted by clock)
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥960 (≥10 days)

Load profile with period 2 (Class ID: 7) **Daily E billing values**			
1	Logical name	Octet-string	**1-0:99.2.0.255**
2	buffer	array	The buffer must be filled monotonously, i.e. no irregular entries are allowed = exactly one entry per capture period

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; {1,0-0:96.10.2.255,2,0}
			{3,0-0:1.8.1.255,2,0}; {3,0-0:1.8.2.255,2,0}
			{3,0-0:2.8.1.255,2,0}; {3,0-0:2.8.2.255,2,0}
			(= clock; AMR profile status; +A rate 1;+A rate 2; –A rate 1;–A rate 2)
			
			Profile status ➔ see paragraph 4.2.5

4	capture_period	double-long-unsigned	86400 (daily)
5	sort_method	enum	1 or 3 (unsorted (FIFO) or sorted (largest))
6	sort_object	object definition	none or {8,0-0:1.0.0.255,2,0} (unsorted or sorted by clock)
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥10 (≥10 days)

6.4 Instantaneous Values (voltage, current, power)

Instantaneous voltage L1 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:32.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V

Average voltage L1 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:32.24.0.255**
2	Value	long-unsigned	10 minutes average voltage. Averaging scheme 3 is used for instantaneous values
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V

Instantaneous current L1 (Class ID: 3)			
1	Logical name	Octet-string	**1-0:31.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A

Instantaneous voltage L2 (Class ID: 3) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:52.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V

Average voltage L2 (Class ID: 3) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:52.24.0.255**
2	Value	long-unsigned	10 minutes average voltage. Averaging scheme 3 is used for instantaneous values
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V

Instantaneous current L2 (Class ID: 3) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:51.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A

Instantaneous voltage L3 (Class ID: 3) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:72.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V

Average voltage L3 (Class ID: 3) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:72.24.0.255**
2	Value	long-unsigned	10 minutes average voltage. Averaging scheme 3 is used for instantaneous values
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V

Instantaneous current L3 (Class ID: 3) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:71.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A

Instantaneous active power (+P) (Class ID: 3)			
1	Logical name	Octet-string	**1-0:15.7.0.255**
2	Value	long-unsigned	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=1, unit=W, resolution: 0.00 kW

Instantaneous current (Class ID: 3)			
1	Logical name	Octet-string	**1-0:90.7.0.255**
2	Value	long-unsigned	Total current (sum of all phases)
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A

6.5 Power Quality (Voltage sags and swells)

Threshold for voltage sag (Class ID: 3)			
1	Logical name	Octet-string	**1-0:12.31.0.255**
2	Value	long-unsigned	Value = 207, threshold for the detection of power sags, programmable according to requirements of the GO
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V

Time threshold for voltage sag (Class ID: 3)			
1	Logical name	Octet-string	**1-0:12.43.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage sag until it is detected, programmable according to requirements of the GO
3	Scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds

Number of voltage sags in phase L1 (Class ID: 1)			
1	Logical name	Octet-string	**1-0:32.32.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage sag until it is detected, programmable according to requirements of the GO

Number of voltage sags in phase L2 (Class ID: 1) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:52.32.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage sag until it is detected, programmable according to requirements of the GO

Number of voltage sags in phase L3 (Class ID: 1) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:72.32.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage sag until it is detected, programmable according to requirements of the GO

Threshold for voltage swell (Class ID: 3)			
1	Logical name	Octet-string	**1-0:12.35.0.255**
2	Value	long-unsigned	Value = 253, threshold for the detection of power swells, programmable according to requirements of the GO
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V

Time threshold for voltage swell (Class ID: 3)			
1	Logical name	Octet-string	**1-0:12.44.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage swell until it is detected, programmable according to requirements of the GO
3	Scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds

Number of voltage swells in phase L1 (Class ID: 1)			
1	Logical name	Octet-string	**1-0:32.36.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage swell until it is detected, programmable according to requirements of the GO

Number of voltage swells in phase L2 (Class ID: 1) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:52.36.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage swell until it is detected, programmable according to requirements of the GO

Number of voltage swells in phase L3 (Class ID: 1) **(polyphase meters only)**			
1	Logical name	Octet-string	**1-0:72.36.0.255**
2	Value	long-unsigned	Value = 30, duration of the voltage swell until it is detected, programmable according to requirements of the GO

7 M-BUS RELATED OBJECTS

7.1 M Bus Master Setup

M-Bus Master Setup (Class ID: 72) **Setup of M-bus master for every M-bus client (4 instances, one per channel, see additional info)**			
1	Logical name	Octet-string	**0-x:24.1.0.255**

The M-Bus Master Setup class is a new class that is not specified in the current version of the Blue Book [1], but it will be defined in the next version. The specification of this class can be found in Appendix A.1 and in [23].

M-Bus port Setup (Class ID: 25)			
1	logical_name	octet-string	0-0:24.0.0.255
2	default_baud	enum	Value = 3; 2400 baud
3	avail_baud	enum	Value = 3; 2400 baud; communication is fixed to 2400 baud
4	addr_state	enum	Value = 1
5	bus_address	unsigned	only relevant if the port also can act as slave

7.2 Identification numbers

Device ID 1 (Class ID: 1) **M-Bus Equipment identifier (4 instances, one per channel)**			
1	Logical name	Octet-string	**0-x:96.1.0.255** (x=channel number (1..4))
2	Value	Octet-string[16]	Mapping to use cases and/or M-Bus unclear

For the COSEM Object Model there are defined M-bus-Identifiers as COSEM Objects:

1. 0.1.96.1.0.255 - CH1 Device ID 1 = CH1 G-Equipment Identifier
2. 0.2.96.1.0.255 - CH2 Device ID 1 = CH2 G-Equipment Identifier
3. 0.3.96.1.0.255 - CH3 Device ID 1 = CH3 G-Equipment Identifier
3. 0.4.96.1.0.255 - CH4 Device ID 1 = CH4 G-Equipment Identifier

Binding is performed with SET to these objects using one of the following methods:

1. Installer sets E-Meter COSEM object with PDA over P0.
2. Installer sets E-Meter COSEM object with selecting from the list of G-Equipment Identifiers on E-Meter.
3. Back office sets E-Meter COSEM object over P3

Once G-Equipment Identifier is in COSEM Object the binding is complete. COSEM objects are defined and this definition is sufficient to perform binding of G-Meter into E-Meter. The method applies for Wired M-Bus and Wireless M-Bus. The difference is that with Wired M-Bus installation is performed with M-Bus Master Setup method install_slave and that COSEM Object is automatically set with G-Equipment Identifier. To release the binding the COSEM Object is cleared.

Device ID 2 (Class ID: 1) **M-Bus Configurator data identifier (4 instances, one per channel)??**			
1	Logical name	Octet-string	**0-x:96.1.1.255** (x=channel number (1..4))
2	Value	Octet-string[48]	Mapping to use cases and/or M-Bus unclear

7.3 Registers

M-Bus Master Value (Class ID: 4) **Instance specific (4 instances, one per channel)**			
1	Logical name	Octet-string	**0-x:24.2.e.255** (x=channel number (1..4), e=instance number (1..4), 16 instances in total: 4 channels with 4 instances per channel)
2	Value	double-long-unsigned	
3	scaler_unit	scal_unit_type	set at installation time in the E-meter
4	status	octet-string	status of M-Bus device
5	capture_time	octet-string	time of last successful readout

7.4 Profiles (hourly)

M-Bus Master Load profile with period 1 (Class ID: 7) **Hourly interval readings of M-Bus devices (4 instances, one per channel)**			
1	Logical name	Octet-string	**0-x:24.3.0.255** (x=channel number (1..4))
2	buffer	array	The buffer must be filled monotonously, i.e. no irregular entries are allowed

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0};
			{1,0-b: 96.10.3.255,2,0}
			{4, 0-b: 24.2.e.255,2,0}
			(=clock; AMR profile status; M-Bus master value object)
			
			AMR profile status see paragraph 4.2.5

4	capture_period	double-long-unsigned	3600, every hour
5	sort_method	enum	1 or 3 (unsorted (FIFO) or sorted (largest))
6	sort_object	object definition	none or {8,0-0:1.0.0.255,2,0}(unsorted or sorted by clock)
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥240 (≥10 days)

7.5 Event Log (Class id = 7)

M-Bus Event Log (Class ID: 7) **M-bus event log containing errors and alarms**			
1	Logical name	Octet-string	**0-0:99.98.3.255**
2	buffer	array	The buffer must be filled monotonously, i.e. no irregular entries are allowed = exactly one entry per capture period

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0};
			{1,0-0.96.11.3.255,2,0}
			(= clock;event code)
			M-bus event codes must be defined, see 4.2.1

4	capture_period	double-long-unsigned	0, asynchronously
5	sort_method	enum	1, unsorted (FIFO)
6	sort_object	object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥10
1	Reset		
2	Capture		Optional, has no meaning

7.6 Disconnector (Control and Log)

M-Bus Master Disconnect Control (Class ID: 70) **Controls the opening and closing of an M-Bus disconnector (e.g. gas valve) (4 instances, one per channel)**			
1	Logical name	Octet-string	**0-x:24.4.0.255** (x=channel number (1..4))
2	output_state	boolean	
3	control_state	enum	
4	control_mode	enum	
1	remote_disconnect		
2	remote_connect		

The Disconnect Control class is a new class that is not specified in the current version of the Blue Book [1], but it will be defined in the next version. The specification of this class can be found in Appendix A.3.

M-Bus Master Control log (Class ID: 7) **Changes of the states related to the disconnect control are recorded (open, close) (4 instances, one per channel)**			
1	Logical name	Octet-string	**0-x:24.5.0.255** (x=channel number (1..4))
2	buffer	Array	

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0};
			{1, 0-b: 96.11.4.255, 2, 0}
			(=clock; control event code)
			
			Event codes must be defined (paragraph 4.2.1)

4	capture_period	double-long-unsigned	0, asynchronously
5	sort_method	Enum	1, unsorted (FIFO)
6	sort_object	Object definition	None, unsorted
7	entries_in_use	double-long-unsigned	
8	profile_entries	double-long-unsigned	≥10

APPENDIX A: SPECIFICATION OF NEW DLMS CLASSES

A.1 M-Bus Master Setup Interface Class (class_id:72)

Instances of this interface class allow setting up and operating devices as M-bus master devices, to exchange data with M-Bus slave devices. An M-Bus master device may have one or more M-Bus interfaces, which can be configured using instances of the M-Bus port setup interface class (class_id: 25). Each M-Bus master setup object controls one M-Bus slave device.

An M-Bus slave device is identified with its Primary Address, Identification Number, Manufacturer ID etc. as defined in EN 13757-3 [20] Clause 5, Variable Data respond. These parameters are carried by the respective attributes of the M-Bus master setup IC.

Values to be captured from an M-Bus slave device are identified by the capture_definition attribute, containing a list of data identifiers (DIB, VIB) for the M-Bus slave device.

Using the methods of M-Bus master setup objects, M-Bus slave devices can be installed and de-installed. Values from a slave device can be captured into M-Bus master value objects and M-Bus master profile generic objects, periodically or on an appropriate trigger. It is also possible to send data to M-Bus slave devices and to perform operations like resetting alarms, setting the clock, controlling disconnect devices (e.g. gas valves).

For details on the M-Bus dedicated application layer, see EN 13757-3 [20].

M-Bus master setup	**0...n**	**class_id = 72, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name (static)	octet-string				x
2. mbus_port_reference (static)	octet-string				x + 0x10
3. capture_definition (static)	array				x + 0x18
4. capture_period (static)	double-long-unsigned				x + 0x20
5. primary_address (dynamic)	unsigned				x + 0x28
6. identification_number (dynamic)	double-long-unsigned				x + 0x30
7. manufacturer_id (dynamic)	long-unsigned				x + 0x38
8. version (dynamic)	unsigned				x + 0x40
9. device_type (dynamic)	unsigned				x + 0x48
10. access_number (dynamic)	unsigned				x + 0x50
11. status (dynamic)	unsigned				x + 0x58
12. alarm (dynamic)	unsigned				x + 0x60
Specific methods (if required)	***m/o***				
1. slave_install	O				x + 0x68
2. slave_deinstall	O				x + 0x70
3. capture	O				x + 0x78
4. reset_alarm	O				x + 0x80
5. synchronize_clock	O				x + 0x88
6. data_send	O				x + 0x90
7. set_encryption_key	O				x + 0x98

Attribute description

logical_name	Identifies the “M-Bus master setup” object instance. For logical name(s), see chapter 7.1.
mbus_port_reference	Provides reference to an M-Bus port setup object, used to configure an M-Bus port, each interface allowing to exchange data with one or more M-Bus slave devices.

capture_definition	Provides the capture_definition for slave devices.
	array capture_definition_element
	capture_definition_element ::= structure
	{
	data_information_block octet-string,
	value_information_block octet-string
	}
	NOTE The elements data_information_block and value_information_block correspond to Data Information Block (DIB) and Value Information Block (VIB) described in EN 13757-3 [20] sub-clause 6.2 and clause 7 respectively.

capture_period	>= 1: Automatic capturing assumed. Specifies the capture period in seconds. 0: No automatic capturing: capturing is triggered externally or capture events occur asynchronously.
primary_address	Carries the primary address of the M-Bus slave device.
identification_number	Carries the Identification Number element of the data header as specified in [20] sub-clause 5.4. It is either a fixed fabrication number or a number changeable by the customer, coded with 8 BCD packed digits (4 Byte), and which thus runs from 00000000 to 99999999. It can be preset at fabrication time with a unique number, but could be changeable afterwards, especially if in addition a unique and not changeable fabrication number (DIF = 0Ch, VIF = 78h, see EN 13757-3 [20] 7.2 is provided.
manufacturer_id	Carries the Manufacturer Identification element of the data header as specified in EN 13757-3 [20] sub-clause 5.5. It is coded unsigned binary with 2 bytes. This manufacturer_id is calculated from the ASCII code of EN 62056-21 manufacturer ID (three uppercase letters), using the formula specified in EN 13757-3 [20] sub-clause 5.5.
version	Carries the Version element of the data header as specified in EN 13757-3 [20] sub-clause 5.6. It specifies the generation or version of the meter and depends on the manufacturer. It can be used to make sure, that within each version number the identification # is unique.
device_type	Carries the Device type identification element of the data header as specified in EN 13757-3 [20] sub-clause 5.7, Table 3.
access_number	Carries the Access Number element of the data header as specified in EN 13757-3 [20] sub-clause 5.8. It has unsigned binary coding, and it is incremented (modulo 256) by one before or after each RSP-UD from the slave. Since it can also be used to enable private end users to detect an unwanted over-frequently readout of its consumption meters, it should not be possible to reset by any bus communication.
status	Carries the Status byte element of the data header as specified in EN 13757-3 [20] sub-clause 5.9, Table 4 and 5.
alarm	Carries the Alarm state specified in EN 13757-3 [20] Annex D. It is coded with data type D (Boolean, in this case 8 bit). Set bits signal alarm bits or alarm codes. The meaning of these bits is manufacturer specific.

Method description

slave_install(data)	Installs slave device data ::= integer (0
slave_deinstall(data)	De-installs slave device data ::= integer (0)
Capture	Capture values from slave device data ::= integer (0)
reset_alarm	Reset alarm state of the slave device data ::= integer (0)

synchronize_clock

Synchronize the clock of the slave device with that of the master device.

```
Data       long-unsigned (0)
```

data_send

Send data to the slave device.

```
data       array data_definition_element
data_definition_element ::= structure
{
           data_information_block      octet-string,
           value_information_block     octet-string
           data
           CHOICE
           {
                     --simple data types
                     null-data                    [0],
                     bit-string                   [4],
                     double-long                            [5],
                     double-long-unsigned                   [6],
                     octet-string                           [9],
                     visible-string                         [10],
                     integer                      [15],
                     long                                   [16],
                     unsigned                     [17],
                     long-unsigned                          [18],
                     long64                                 [20],
                     long64-unsigned              [21],
                     float32                      [23],
                     float64                      [24]
           }
}
```

The following objects are available to configure M-Bus master devices and to exchange data with M-Bus slave devices:

- instances of the M-Bus master setup IC are used to configure devices as M-Bus masters;
- values captured from M-Bus slave devices are held by M-Bus masters value objects or M-Bus master profile generic objects;
- disconnector devices of M-Bus master devices (e.g. gas valves) are controlled by instances of the M-Bus master disconnect control IC;
- changes of the state of the disconnect device are held by instances of M-Bus master control log IC.

M-Bus master	IC	OBIS code					
		A	B	C	D	E	F
M-Bus master setup objects	72, M-Bus master setup	0	b	24	1	0	255
M-Bus master value objects	4, Extended register	0	b	24	2	e [a]	255
M-Bus master profile generic objects	7, Profile generic	0	b	24	3	0	255
M-Bus master disconnect control objects	70, Disconnect control	0	b	24	4	0	255
M-Bus master control log objects	7, Profile generic	0	b	24	5	0	255

[a] "e" is equal to the index of the captured value in accordance to index of capture_definition_element in the capture_definition attribute of the MBUS master setup object.

A.2 Limiter Interface Class (class_id:71)

Instances of the Limiter interface class allow defining a set of actions that are executed when the value of a value attribute of a monitored object “Data”, “Register”, “Extended Register”, “Demand Register”, etc. crosses the threshold value for at least minimal duration time.

The threshold value can be normal or emergency threshold. The emergency threshold is activated via the emergency profile defined by emergency profile id, activation start time, and duration. The emergency profile id element is matched to an emergency profile group id: this mechanism enables the activation of the emergency threshold only for a specific emergency group.

Limiter		**0...n**	**class_id = 71, version = 0**			
Attribute(s)		***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name	(static)	octet-string				x
2. monitored_value	(static)	value_definition_ type				x + 0x08
3. threshold_active	(dynamic)	threshold				x + 0x10
4. threshold_normal	(static)	threshold				x + 0x18
5. threshold_emergency	(static)	threshold				x + 0x20
6. min_over_threshold_duration	(static)	double-long-unsigned				x + 0x28
7. min_under_threshold_duration	(static)	double-long-unsigned				x + 0x30
8. emergency_profile	(static)	emergency_profile _type				x + 0x38
9. emergency_profile_group _id_list	(static)	array				x + 0x40
10. emergency_profile_active	(dynamic)	boolean				x + 0x48
11. actions	(static)	action_type				x + 0x50
Specific methods (if required)		***m/o***				

Attribute description

logical_name

Identifies the “Limiter” object instance. For logical name(s), see chapter 6.7.

monitored_value

Defines an attribute of an object to be monitored. Only attributes with simple data types are allowed.

```
value_definition_type ::= structure
{
            class_id: long-unsigned,
            logical_name: octet-string,
            attribute_index: integer
}
```

threshold_active

Provides the active threshold value to which the attribute monitored is compared.

data   threshold

threshold:The threshold is of the same type as the attribute monitored

threshold_normal

Provides the threshold value to which the attribute monitored is compared when in normal operation.

data   threshold

threshold:The threshold is of the same type as the attribute monitored.

threshold_emergency

Provides the threshold value to which the attribute monitored is compared when an emergency profile is active.

```
data                    threshold
```

threshold: The threshold is of the same type as the attribute monitored.

min_over_threshold_duration

Defines minimal over threshold duration in seconds required to execute the over threshold action.

min_under_threshold_duration

Defines minimal under threshold duration in seconds required to execute the under threshold action.

emergency_profile

An emergency_profile is defined by three elements: emergency_profile_id, emergency_activation_time and emergency_duration.

An emergency profile is activated if the emergency_profile_id element matches one of the elements on the emergency_profile _group_id_list, and time matches the emergency_activation_time and emergency_duration element:

```
emergency_profile_type ::= structure
{
        emergency_profile_id:               long-unsigned,
        emergency_activation_time:  octet-string,
        emergency_duration:                 double-long-unsigned
}
```

emergency_activation_time defines the date and time when the emergency_profile activated. The octet-string is encoded as specified in [1] 4.1.6.1 for date_time.

emergency_duration defines the duration in seconds, for which the emergency_profile is activated.

When an emergency profile is active, the emergency_profile_active attribute is set to TRUE.

emergency_profile_group _id_list

Defines a list of group id-s of the emergency profile.

The emergency profile can be activated only if emergency_profile_id element of the emergency_profile_type matches one of the elements on the emergency_profile _group_id-list:

```
array                   emergency_profile_group_id
```

emergency_profile_group_id: long-unsigned.

emergency_profile_active

Indicates that the emergency_profile is active.

Actions

Defines the scripts to be executed when the monitored value crosses the threshold for minimal duration time.

```
action_type ::= structure
{
        action_over_threshold: action_item;
        action_under_threshold: action_item
}
```

where:

- action_over_threshold defines the action when the value of the attribute monitored crosses the threshold in upwards direction and remains over threshold for minimal over threshold duration time;
- action_under_threshold defines the action when the value of the attribute monitored crosses the threshold in the downwards direction and remains under threshold for minimal under threshold duration time.

```
action_item::= structure
{
        script_logical_name: octet-string,
        script_selector: long-unsigned
}
```

A.3 Disconnect control (class_id:70)

Instances of the Disconnect control interface class manage an internal or external disconnect unit of the meter (e.g. electricity breaker, gas valve) in order to connect or disconnect, partly or entirely, the premises of the consumer.

The state diagram and the possible state transitions are shown in figure A.3.1

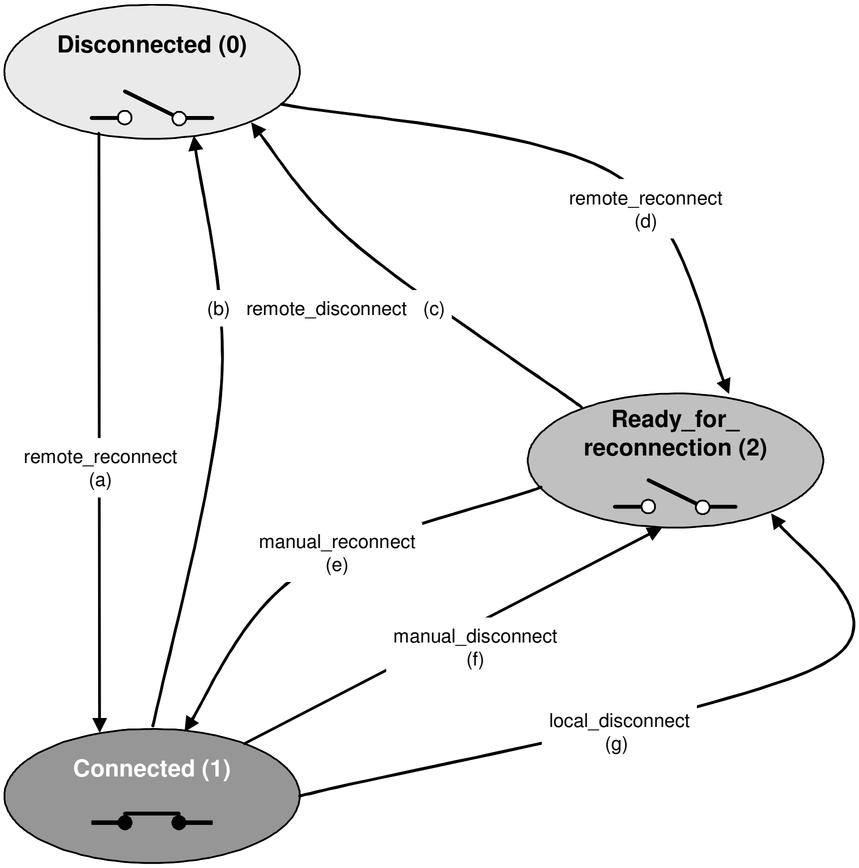


Figure A.3.1 – State diagram of the Disconnect control interface class

Disconnect and reconnect can be requested:

- Remotely, via a communication channel: remote_disconnect, remote_reconnect;
- Manually, using e.g. a push button: manual_disconnect, manual_reconnect;
- Locally, by a function of the meter, e.g. limiter, prepayment: local_disconnect. Local reconnection is not possible: reconnection after a local_disconnect always requires a manual intervention.

The possible states and state transitions of the Disconnect control interface class are shown in table A.3.1.The Disconnect control object doesn't feature a memory, i.e. any commands are executed immediately.

Table A.3.1 – Disconnect control IC – states and state transitions

States		
State number	**State name**	**State description**
0	Disconnected	The output_state is set to FALSE and the consumer is disconnected.
1	Connected	The output_state is set to TRUE and the consumer is connected.
2	Ready_for_reconnection	The output_state is set to FALSE and the consumer is disconnected. Reconnection requires manual intervention.
State transitions		
Transition	**Transition name**	**State description**
a	remote_reconnect	Moves the Disconnect control object from the Disconnected (0) state directly to the Connected (1) state without manual intervention
b	remote_disconnect	Moves the Disconnect control object from the Connected (1) state to the Disconnected (0) state
c	remote_disconnect	Moves the Disconnect control object from the Ready_for_reconnection (2) state to the Disconnected (0) state
d	remote_reconnect	Moves the Disconnect control object from the Disconnected (0) state to the Ready_for_reconnection (2) state From this state, it is possible to move to the Connected (2) state via the manual_reconnect transition (e)
e	manual_reconnect	Moves the Disconnect control object from the Ready_for_connection (2) state to the Connected (1) state
f	manual_disconnect	Moves the Disconnect control object from the Connected (1) state to the Ready_for_connection (2) state From this state, it is possible to move back to the Connected (2) state via the manual_reconnect transition (e)
g	local_disconnect	Moves the Disconnect control object from the Connected (1) state to the Ready_for_connection (2) state From this state, it is possible to move back to the Connected (2) state via the manual_reconnect transition (e)NOTE Transitions f) and g) are essentially the same, but their trigger is different.

To define the behaviour of the disconnect control object for each trigger, the control mode must be set.

Disconnect control	**0...n**	**class_id = 70, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	***Short name***
1. logical_name (static)	octet-string				x
2. output_state (dyn.)	boolean				x + 0x08
3. control_state (dyn.)	enum				x + 0x10
4. control_mode (static)	enum				x + 0x18
Specific methods	**m/o**				
1. remote_disconnect()	m				x + 0x20
2. remote_reconnect()	m				x + 0x28

Attribute description

logical_name	Identifies the “Disconnect control” object instance. For logical name(s) see chapter 6.6 and 7.6.

output_state	Shows the actual physical state of the disconnect unit, i.e. if an electricity breaker or a gas valve is open or closed.
	
	boolean
	TRUE = connected,
	FALSE = disconnected

control_state	Shows the internal state of the disconnect control object. enum (0) = Disconnected (1) = Connected (2) = Ready_for_reconnection
control_mode	Configures the behaviour of the disconnect control object for all triggers. enum Possible state transitions (0) None. The disconnect control object is always in 'connected' state (1) Disconnection: Remote (b, c), manual (f), local (g) Reconnection: Remote (d), manual (e) (2) Disconnection: Remote (b, c), manual (f), local (g) Reconnection: Remote (a), manual (e) (3) Disconnection: Remote (b, c), manual (-), local (g) Reconnection: Remote (d), manual (e) (4) Disconnection: Remote (b, c), manual (-), local (g) Reconnection: Remote (a), manual (e) NOTE Local disconnection is always possible. To suppress local disconnection, the corresponding trigger must be inhibited.

Method description

remote_disconnect ()	Forces the disconnect control object into 'disconnected' state if remote disconnection is enabled (control mode > 0).
remote_reconnect ()	Forces the disconnect control object into the 'ready_for_reconnection' state if a direct remote reconnection is disabled (control mode = 1, 3). Forces the disconnect control object into the 'connected' state if a direct remote reconnection is enabled (control mode = 2, 4).

APPENDIX B: PLC SETUP CLASSES

B.1 S-FSK Phy&MAC setup (class_id: 50)

An instance of the “S-FSK Phy&MAC setup” class stores the data necessary to set up and manage the physical and the MAC layer of the PLC S-FSK lower layer profile.

S-FSK Phy&MAC setup	**0...n**	**class_id = 50, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. initiator_electrical_phase (static)	unsigned	0	2		x + 0x08
3. delta_electrical_phase (dyn.)	enum	0	6		x + 0x10
4. max_receiving_gain (static)	unsigned				x + 0x18
5. max_transmitting_gain (static)	unsigned				x + 0x20
6. search_initiator_gain (static)	unsigned				x + 0x28
7. frequencies (static)	frequencies_type				x + 0x30
8. mac_address (dyn.)	long-unsigned			FFE	x + 0x38
9. mac_group_addresses (static)	array				x + 0x40
10. repeater (static)	enum			1	x + 0x48
11. repeater_status (dyn.)	boolean				x + 0x50
12. min_delta_credit (dyn)	unsigned				X + 0x58
13. initiator_mac_address (dyn.)	long-unsigned				x + 0x60
14. synchronization_locked (dyn.)	boolean				x + 0x68
Specific methods	***m/o***				

Attribute description

logical_name Identifies the “S-FSK Phy&MAC setup” object instance.

initiator_electrical_phase Holds the MIB variable *initiator-electrical-phase* (variable 18) specified in IEC 61334-4-512 sub-clause 5.8.

It is written by the client system to indicate the phase to which it is connected.

delta_electrical_phase Holds the MIB variable *delta-electrical-phase* (variable 1) specified in IEC 61334-4-512 sub-clause 5.2 and IEC 61334-5-1 sub-clause 3.5.5.3.

It indicates the phase difference between the client's connecting phase and the server's connecting phase. The following values are predefined:

0: Not defined: the server is temporarily not able to determine the phase difference;

1: the server system is connected to the same phase as the client system;

2: the phase difference between the server's connecting phase and the client's connecting phase is equal to 60 degrees;

3: the phase difference between the server's connecting phase and the client's connecting phase is equal to 120 degrees;

4: the phase difference between the server's connecting phase and the client's connecting phase is equal to 180 degrees;

5: the phase difference between the server's connecting phase and the client's connecting phase is equal to -120 degrees;

6: the phase difference between the server's connecting phase and the client's connecting phase is equal to –60 degrees.

max_receiving_gain

Holds the MIB variable *max-receiving-gain* (variable 2) specified in IEC 61334-4-512 sub-clause 5.2 and IEC 61334-5-1 sub-clause 3.5.5.3.

Corresponds to the maximum allowed gain bound to be used by the server system in the receiving mode. The default unit is dB.

NOTE 1 In IEC 61334-5-1, no units is specified.

NOTE 2 The possible values of the gain may depend on the hardware. Therefore, after writing a value to this attribute, the value should be read back to know the actual value.

max_transmitting_gain

Holds the value of the max-transmitting-gain.

Corresponds to the maximum attenuation bound to be used by the server system in the transmitting mode. The default unit is dB.

NOTE 1 In IEC 61334-4-512, no units is specified.

NOTE 2 The possible values of the gain may depend on the hardware. Therefore, after writing a value to this attribute, the value should be read back to know the actual value.

search_initiator_gain

This attribute is used in the intelligent search initiator process. If the value of the max_receiving_gain is below the value of this attribute, a fast synchronization process is possible.

Frequencies

Contains frequencies required for S-FSK modulation.

```
frequencies_type          structure
{
            mark_frequency    double-long-unsigned,
            space_frequency   double-long-unsigned
}
```

The default unit is Hz.

mac_address

Holds the MIB variable *mac-address* (variable 3) specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 sub-clause 4.3.7.6.

NOTE MAC addresses are expressed on 12 bits.

Contains the value of the physical attachment (MAC address) associated to the local system. In the unconfigured state, the MAC address is “NEW-address”.

This attribute is locally written by the CIASE when the system is registered (with a Register service). The value is used in each outgoing or incoming frame. The default value is "NEW-address".

This attribute is set to NEW:

- by the MAC sub-layer, once the time-out-not-addressed delay is exceeded;
- when a client system “resets” the server system. See the “S-FSK Active initiator” IC.

When this attribute is set to NEW:

- the system loses its synchronisation (function of the MAC-sublayer);
- the mac_group_address attribute is reset (array of 0 elements);
- the system automatically releases all AAs which can be released.

NOTE The second item is not present in IEC 61334-4-512

The predefined MAC addresses are shown in Table B.1.1.

mac_group_address

Holds the MIB variable *mac-group-address* (variable 4) specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 sub-clause 4.3.7.6.

Contains a set of MAC group addresses used for broadcast purposes.

array	mac-address
mac-address	long-unsigned

The ALL-configured-address, ALL-physical-address and NO-BODY addresses are not included in this list. These ones are internal predefined values. This attribute shall be written by the initiator using DLMS services to declare specific MAC group addresses on a server system.

This attribute is locally read by the MAC sublayer when checking the destination address field of a MAC frame not recognized as an individual address or as one of the three predefined values (ALL-configured-address, ALL-physical-address and NO-BODY).

repeater	Holds the MIB variable *repeater* (variable 5) specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 sub-clause 4.3.7.6. Holds the information whether the server system effectively repeats all frames or not. enum (0) never repeater (1) always repeater (2) dynamic repeater If the repeater variable is equal to 0, the server system should never repeat the frames. If it is set to 1, the server system is a repeater: it has to repeat all frames received without error and with a current credit greater than zero. If it is set to 2, then the repeater status can be dynamically changed by the server itself. NOTE The value 2 value is not specified in IEC 61334-4-512. This attribute is internally read by the MAC sub-layer each time a frame is received. The default value is 1.

repeater_status	Holds the current repeater status of the device.
	Boolean
	FALSE = no repeater
	TRUE = repeater

min_delta_credit	Holds the MIB variable *min-delta-credit* (variable 9) specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 4.3.7.6. NOTE Only the three least significant bits are used. The Delta Credit (DC) is the subtraction of the Initial Credit (IC) and Current Credit (CC) fields of a correct received MAC frame. The delta-credit minimum value of a correct received MAC frame, directed to a server system, is stored in this attribute. The default value is set to the maximal initial credit (see IEC 61334-5-1 4.2.3.1 [6] for further explanations on the credit and the value of MAX_INITIAL_CREDIT). A client system can reinitialize this variable by setting its value to the maximal initial credit.
initiator_mac_address	Holds the MIB variable *initiator-mac-address* specified in IEC 61334-5-1 4.3.7.6. Its value is either the MAC address of the active-initiator or the NO-BODY address, depending on the value of the synchronisation_locked attribute (see below). See also IEC 61334-5-1 3.5.3, 4.1.6.3 and 4.1.7.2. NOTE If the value NO-BODY is written then the server mac_address (see the mac_address attribute) has to be set to NEW.
synchronisation_locked	Holds the MIB variable *synchronization-locked* (variable 10) specified in IEC 61334-4-512 sub-clause 5.3. Controls the synchronisation locked / unlocked state. See in IEC 61334-5-1 for more details. If the value of this attribute is equal to TRUE, the system is in the synchronisation-locked state. In this state, the *initiator-mac-address* is always equal to the MAC address field of the *active-initiator* MIB object. See attribute 2 of the S-FSK Active initiator IC. If the value of this attribute is equal to FALSE, the system is in the synchronisation-unlocked state. In this state, the *initiator_mac_address* attribute is always set to the NO-BODY value: a value change in the MAC address field of the *active-initiator* MIB object does not affect the content of the *initiator_mac_address* attribute which remains at the NO-BODY value. The default value of this variable shall be specified in the implementation specifications. NOTE In the synchronisation-unlocked state, the server synchronises on any valid frame. In the synchronisation locked state, the server only synchronises on frames issued or directed to the client system the MAC address of which is equal to the value of the initiator_mac_address attribute.

Table B.1.1 – MAC addresses in the S-FSK profile

Address	Value
NO-BODY	000
Local MAC	001…FIMA-1
Initiator	FIMA…LIMA
MAC group address	LIMA + 1…FFB
All-configured	FFC
NEW	FFE
All Physical	FFF

NOTE MAC addresses are expressed on 12 bits. These addresses are specified in IEC 61334-5-1 sub-clauses 4.2.3.2, 4.3.7.5.1, 4.3.7.5.2 and 4.3.7.5.3.
FIMA = First Initiator MAC address; C00
LIMA = Last Initiator MAC address; DFF	

B.2 S-FSK Active initiator (class_id: 51)

An instance of the "S-FSK Active initiator" class stores the data of the active initiator. The active initiator is the client system, which has last registered the server system with a CIASE Register request. See IEC 61334-5-511 7.2.

S-FSK Active initiator		**0...n**	**class_id = 51, version = 0**			
Attribute(s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. active_initiator	(dyn.)	initiator_descriptor				x + 0x08
Specific methods		***m/o***				
1. reset_NEW_not_synchronised						x + 0x10

Attribute description

logical_name	Identifies the "S-FSK Active initiator" object instance

active_initiator	Holds the MIB variable *active-initiator* (variable 15) specified in IEC 61334-4-512 sub-clause 5.6.
	Contains the identifiers of the active initiator which has last registered the system with a Register request. See IEC 61334-4-511 7.2.
	The Initiator system is identified with its System Title, MAC address and L-SAP selector:
	initiator-descriptor ::=structure
	{
	system_title octet-string (SIZE(SYSTEM-TITLE-SIZE)),
	MAC_address long-unsigned,
	L-SAP_selector unsigned
	}
	Unless otherwise specified in a system specification:
	• When the IEC 61334-4-32 LLC layer is used, SYSTEM-TITLE-SIZE = 6;
	• When the HDLC based data link layer is used (IEC 62056-46, Green Book Clause 8),

SYSTEM-TITLE-SIZE = 8;

- When the ISO/IEC 8802-2 LLC layer is used, SYSTEM-TITLE-SIZE = 16.

NOTE The system title of the initiator should have the same structure as the COSEM logical device name specified in DLMS UA 1000-1 sub-clause 4.1.8.2.

If the System-Title field of this object is set to an octet string of 0s, it means that the system is not registered.

The MAC-address element is used to update the *initiator-mac-address* MAC management variable when the system is configured in the synchronisation-locked state. See the specification of the *initiator_mac_address* and the *synchronisation_locked* attributes of the S-FSK Phy&MAC setup IC.

As long as the server is not registered by an active initiator, the LSAP_selector field is set to 0 and the system_title field is equal to an octet string of 0s.

The default value of the initiator-descriptor is: system_title = octet-string of 0s, MAC_address = NO-BODY and L-SAP_selector = 0.

The value of this attribute can be updated by the invocation of the reset_NEW_not_synchronised method or by the CIASE Register service.

Method description

reset_NEW_not_ synchronised (data)

Holds the MIB variable *reset-NEW-not-synchronised* (variable 17) specified in IEC 61334-4-512 sub-clause 5.8.

Allows a client system to “reset” the server system. The submitted value corresponds to a client MAC address. The writing is refused if the value does not correspond to a valid client MAC address or the predefined NO-BODY address.

When this method is invoked, the following actions are performed:

- the system returns to the unconfigured state (UNC: MAC-address equals NEW-address). This transition automatically causes the synchronisation lost (function of the MAC sub layer);
- the system changes the value of the *active_initiator* attribute: the MAC address is set to the submitted value, the LSAP selector is set to the value 0 and the System-Title is set to an octet-string of 0s.
- all AAs that can be released are released.

B.3 S-FSK MAC synchronisation timeouts (class_id: 52)

An instance of the “S-FSK synchronisation timeouts” class stores the timeouts related to the synchronization process.

S-FSK MAC synchronisation timeouts	**0...n**	**class_id = 52, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. search_initiator_timeout (static)	long-unsigned				x + 0x08
3. synchronisation_confirmation_ timeout (static)	long-unsigned				x + 0x10
4. timeout_not_addressed (static)	long-unsigned				x + 0x18
5. timeout_frame_not_OK (static)	long-unsigned				x + 0x20
Specific methods	***m/o***				

Attribute description

logical_name	Identifies the “S-FSK synchronisation timeouts” object instance.
search_initiator_ timeout	This timeout supports the intelligent search initiator function. It defines the value of the time, expressed in seconds, during which the server system is searching for the initiator with the strongest signal. During this timeout, all initiators, which may be heard by the servers, are expected to talk. After the expiry of this timeout, the server will accept a Register request from the initiator having provided the strongest signal and it will be locked to that initiator. If the value of the timeout is equal to 0, this means that the feature is not used. The timeout starts when the server receives the first frame with a valid initiator MAC address. NOTE A Fast synchronization may be performed if the level of signal and the gain are good enough (Gain <= Search-Initiator-Gain) and one of the MAC addresses (Source or Destination) is an Initiator MAC address. This means the module is next to a DC or next to a module that is already locked on that DC. The module locks in this case on that initiator.
synchronisation_ confirmation_timeout	Holds the MIB variable *synchronisation-confirmation-timeout* (variable 6*)* specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 4.3.7.6. Defines the value of the time, expressed in seconds, after which a server system which just gets frame synchronised (detection of a data path equal to AAAA54C7 hex) will automatically lose its frame synchronisation if the MAC sublayer does not identify a valid MAC frame. The timeout starts after the reception of the first four bytes of a physical frame. The value of this variable can be modified by a client system. This time-out ensures a fast desynchronisation of a system, which has synchronised on a wrong physical frame. See IEC 61334-5-1 3.5.3 for more details. NOTE The default value of this variable should be specified in the implementation specifications. A value equal to 0 is equivalent to cancel the use of the related synchronisation_confirmation_timeout counter.
timeout_not_addressed	Holds the MIB variable *timeout-not-addressed* (variable 7) specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 4.3.7.6. Defines the time, in minutes, after which a server system that has not been individually addressed: - returns to the non configured state (UNC: MAC-address equals NEW-address): this transition automatically involves the loss of the synchronisation (function of the MAC sub layer) and releasing all AAs that can be released; - loses its active initiator: the MAC address of the active-initiator is set to NO-BODY, the LSAP selector is set to the value 00 and the System Title is set to an octet-string of 0s. Because broadcast addresses are not individual system addresses, the timer associated with the *time-out-not-addressed* delay ensures that a forgotten system will sooner or later return to the unconfigured state. It will be then discovered again. A forgotten system is a system, which has not been individually addressed for more than the "*time-out-not-addressed*" amount of time. NOTE The default value of this variable should be specified in the implementation specifications. A value equal to 0 is equivalent to cancel the use of the related *time-out-not-addressed* counter.

timeout_frame_not_OK	Holds the MIB variable *time-out-frame-not-OK* (variable 8), specified in IEC 61334-4-512 sub-clause 5.3 and IEC 61334-5-1 4.3.7.6. Defines the time, in seconds, after which a server system that has not received a properly formed MAC frame (incorrect NS field, inconsistent number of received sub frames, false Cyclic Redundancy Code checking) loses its frame synchronisation. The default value of this variable shall be specified in the implementation specifications. A value equal to 0 is equivalent to cancel the use of the related *time-out-frame-not-OK* counter.

B.4 S-FSK MAC counters (class_id: 53)

An instance of the “S-FSK counters” class stores counters related to the frame exchange, transmission and repetition phases.

S-FSK MAC counters		**0...n**	**class_id = 53, version = 0**			
Attribute(s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				X
2. synchronisation_register	(dyn.)	array				x + 0x08
3. desynchronisation_listing	(dyn.)	structure				x + 0x10
4. broadcast_frames_counter	(dyn.)	array				x + 0x18
5. repetitions_counter	(dyn.)	double-long-unsigned			0	x + 0x20
6. transmissions_counter	(dyn.)	double-long-unsigned			0	x + 0x28
7. CRC_OK_frames_counter	(dyn.)	double-long-unsigned			0	x + 0x30
8. CRC_NOK_frames_counter	(dyn.)	double-long-unsigned				x + 0x38
Specific methods		***m/o***				

Attribute description

synchronisation_ register

Holds the MIB variable *synchronisation-register* (variable 23), specified in IEC 61334-4-512 sub-clause 5.8.

```
array                              synchronization-couples

synchronization-couples ::=        structure
        {
              mac-address                        long-unsigned,
              synchronisations-counter           double-long-unsigned
        }
```

This variable counts the number of synchronisation processes performed by the system. Processes that lead to a synchronisation loss due to the detection of a wrong initiator are registered. The other processes that lead to a synchronisation loss (time-out, Management Writing) **are not** registered. This variable provides a balance sheet of the different systems on which the server system is "potentially" able to synchronise.

A synchronisation process is initialized when the Management Application Entity (connection manager) receives a MA_Sync.indication (Synchronisation State = SYNCHRO_FOUND) primitive from the MAC Sublayer Entity. This process is registered in the *synchronisation-register* variable only if the MA_Sync.indication (Synchronisation State = SYNCHRO_FOUND) primitive is followed by one of the three primitives:

1) MA_Data.indication(DA, SA, M_sdu) primitive;
2) MA_Sync.indication (Synchronisation State = SYNCHRO_CONF, SA, DA) ;
3) MA_Sync.indication(Synchronisation State = SYNCHRO_LOSS, Synchro Loss

Cause = wrong_initiator, SA, DA).

NOTE The third primitive is only generated if the server system is configured in a synchronisation-locked state. See Appendix B.1.

Processes which lead to the generation of MA_Sync.indication (Synchronisation State = SYNCHRO_LOSS) primitives indicating synchronisation loss due to:

- the physical layer;
- the time-out-not-addressed counter;
- setting the mac_address attribute of the S-FSK Phy&MAC setup object to NEW; see Appendix B.1; or invoking the reset_NEW_not_synchronized attribute of the S-FSK Active initiator object; see Appendix B.2 (this is known as Management Writing)

are not taken into account in this variable.

For details on the MA_Sync.indication service primitive see IEC 61334-5-1 sub-clause 4.1.7.1.

If the synchronisation process ends with one of the three primitives listed above, the *synchronisation-register* variable is updated by taking into account the SA and DA fields of the primitive.

The updating of the *synchronisation-register* variable is carried out as follows:

synchronisation_ register (continued)

First, the Management Entity checks the SA and DA fields.

- If one of these fields corresponds to a client MAC address (CMA) the Entity:
  - ○ checks if the client MAC address (CMA) appears in one of the couples contained in the *synchronisation-register* variable;
  - ○ If it appears, the related synchronizations-counter subfield is incremented;
  - ○ If it does not appear, a new (mac-address, synchronizations-counter) couple is added. This couple is initialized to the (CMA, 1) value.
- If none of the SA and DA fields correspond to a client MAC address, it is supposed that the system found its synchronisation reference on a DiscoverReport type frame. In that case, the mac-address which should be registered in the synchronization-register variable is the predefined NEW value (0FFE). The updating of the synchronisation-register variable is carried out in the same way as it is done for a normal client MAC address (CMA).

When a *synchronisations-counter* field reaches the maximum value, it automatically returns to 0 on the next increment.

The maximum number of synchronisation couples {mac-address, synchronisations-counter} contained in this variable should be specified in the implementation specifications. When this maximum is reached, the updating of the variable follows a First-In-First-Out (FIFO) mechanism: only the newest source MAC addresses are memorized.

The default value of this variable is an empty array.

desynchronisation-listing

Holds the MIB variable *desynchronisation-listing* (variable 24), specified in IEC 61334-4-512 sub-clause 5.8.

```
desynchronisation_listing ::= structure
{
    nb_physical_layer_desynchronisation             double_long_unsigned;
    nb_time_out_not_addressed_desynchronisation     double_long_unsigned;
    nb_timeout_frame_not_OK desynchronisation       double_long_unsigned;
    nb_write_request_desynchronisation              double_long_unsigned;
    nb_wrong_initiator_desynchronisation            double_long_unsigned;
}
```

This variable counts the number of desynchronisations that occurred depending on their cause. On reception of synchronisation loss notification, the Management Entity updates this attribute by incrementing the counter related to the cause of the desynchronisation.

When one of the counters reaches the maximum value, it automatically returns to 0 on the next increment.

The default value of this variable contains elements which are all equal to 0.

broadcast_frames_ counter

Holds the MIB variable *broadcast-frames-counter* (variable 19) specified in IEC 61334-4-512 sub-clause 5.8.

array broadcast-couples

```
broadcast-couples   structure
          {
                    source-mac-address          long-unsigned,
                    frames-counter              double-long-unsigned
          }
```

It counts the broadcast frames received by the server system and issued from a client system (source-mac-address = any valid client-mac-address, destination-mac address = ALL-physical). The number of frames is classified according to the origin of the transmitter. The counter is incremented even if the LLC-destination-address is not valid on the server system. When the frames-counter field reaches its maximum value, it automatically returns to 0 on the next increment. The maximum number of broadcast-couples {source-mac-address, frames-counter} contained in this variable should be specified in the implementation specifications. When this maximum is reached, the updating of the variable follows a First-In- First-Out (FIFO) mechanism: only the newest source MAC addresses are memorized.

repetitions_counter

Holds the MIB variable *repetitions-counter* (variable 20) specified in IEC 61334-4-512 sub-clause 5.8.

Counts the number of repetition phases. The repetition phases following a transmission are not counted. If the MAC sub-layer is configured in the no-repeater mode, this variable is not updated. The *repetitions-counter* measures the activity of the system as a repeater. A received frame repeated five times (from CC=4 to CC=0) is counted only once in the *repetitions-counter* since it corresponds to one repetition phase. The counter is incremented at the beginning of each repetition phase. When the *repetitions-counter* reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

transmissions_counter

Holds the MIB variable *transmissions-counter* (variable 21) specified in IEC 61334-4-512 sub-clause 5.8

Counts the number of transmission phases. A transmission phase is characterized by the transmission and the repetition of a frame. A repetition phase, which follows the reception of a frame is not counted. The transmission counter is incremented at the beginning of each transmission phase. A client system can write this variable to update the counter. When the transmissions-counter reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

CRC_OK_frames_ counter

Holds the MIB variable *CRC-OK-frames-counter* (variable 22) specified in IEC 61334-4-512 sub-clause 5.8

Counts the number of frames received with a correct Frame Check Sequence Field. When the CRC-OK-frames counter field reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

CRC_NOK_frames_ counter

Counts the number of frames received with an incorrect Frame Check Sequence Field. When the CRC-NOK-frames counter field reaches the maximum value, it automatically returns to 0 on the next increment. The default value is 0.

B.5 S-FSK IEC 61334-4-32 LLC setup (class_id: 55)

An instance of the "S-FSK IEC 61334-4-32 LLC setup "class holds parameters necessary to set up and manage the LLC layer as specified in IEC 61334-4-32.

S-FSK IEC 61334-4-32 LLC setup	**0...n**	**class_id = 55, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. max_frame_length (static)	unsigned	26	242	134	x + 0x08
3. reply_status_list (dyn.)	array				x + 0x10
Specific methods	***m/o***				

Attribute description

logical_name	Identifies the “S-FSK IEC 61334-4-32 LLC setup” object instance.
max_frame_length	Holds the length of the LLC frame in bytes. See IEC 61334-4-32 sub-clause 5.1.4. In the case of the S-FSK profile, as specified in 61334-5-1 sub-clause 4.2.2, the maximum value is 242, but lower values may be chosen due to performance considerations.

reply_status_list	Holds the MIB variable *reply-status-list* (variable 11) specified in 61334-4-512 sub-clause 5.4.
	Lists the L-SAPs that have a not empty RDR (Reply Data on Request) buffer, which has not already been read. The length of a waiting L-SDU is specified in number of sub frames (different from zero). The variable is locally generated by the LLC sub layer.
	reply_status_list ::= array of reply_status
	reply_status ::= structure
	{
	L-SAP-selector unsigned,
	length-of-waiting-L-SDU unsigned
	}
	length-of-waiting-LSDU in the case of the S-FSK profile is in number of sub-frames; valid values are 1 to 7.

B.6 S-FSK Reporting system list (class_id: 56)

An instance of the “S-FSK Reporting system list” class holds the list of reporting systems.

S-FSK Reporting system list	**0...n**	**class_id = 56, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. reporting_system_list (dyn.)	array				x + 0x08
Specific methods	***m/o***				

Attribute description

logical_name

Identifies the “S-FSK Reporting system list” object instance.

reporting_system_list

Holds the MIB variable *reporting-system-list* (variable 16) specified in 61334-4-512 sub-clause 5.7.

array	system-title
system-title	octet-string

Contains the system-titles of the server systems which have made a DiscoverReport request and which have not already been registered. The list has a finite size and it is sorted upon the arrival. The first element is the newest one. Once full, the oldest ones are replaced by the new ones.

The reporting-system-list is updated:

- when a DiscoverReport CI_PDU is received by the server system (whatever its state: non configured or configured): the CIASE adds the reporting system-title at the beginning of the list, and verifies that it does not exist anywhere else in the list, if so it destroys the old one. A system-title can only be present once in the list;
- when a Register CI_PDU is received by the server system (whatever its state: non configured or configured): the CIASE checks the reporting-system list. If a system-title is present in the reporting-system-list and in the Register CI-PDU, the CIASE deletes the system-title in the reporting-system-list: this system is no more considered as a reporting system.

The number of elements in the array is limited to one.

B.7 ISO/IEC 8802-2 LLC Type 1 setup (class_id: 57)

An instance of the ISO/IEC 8802-2 LLC Type 1 setup class holds the parameters necessary to set up the ISO/IEC 8802-2 LLC layer in Type 1 operation.

ISO/IEC 8802-2 LLC Type 1 setup		0...n	class_id = 57, version = 0			
Attribute(s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				X
2. max_octets_ui_pdu	(static)	long unsigned			128	x + 0x08
Specific methods		***m/o***				

Attribute description

logical_name

Identifies the “ISO/IEC 8802-2 LLC Type 1 setup” object instance.

max_octets_ui_pdu

Refer to the appropriate MAC protocol specification for any limitation on the maximum number of octets in a UI PDU. No restrictions are imposed by the LLC sublayer. However, in the interest of having a value that all users of Type 1 LLC may depend upon, all MACs must at least be capable of accommodating UI PDUs with information fields up to and including 128 octets in length.

See ISO/IEC 8802-2 sub-clause 6.8.1 *Maximum number of octets in a UI PDU.*

B.8 ISO/IEC 8802-2 LLC Type 2 setup (class_id: 58)

An instance of the ISO/IEC 8802-2 LLC Type 2 setup class holds the parameters necessary to set up the ISO/IEC 8802-2 LLC layer in Type 2 operation.

ISO/IEC 8802-2 LLC Type 2 setup	0...n	class_id = 58, version = 0			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. transmit_window_size_k (static)	unsigned	1	127	1	x + 0x08
3. receive_window_size_rw (static)	unsigned	1	127	1	x + 0x10
4. max_octets_i_pdu_n1 (static)	long unsigned			128	x + 0x18
5. max_number_transmissions_n2 (static)	unsigned				x + 0x20
6. acknowledgement_timer (static)	long unsigned				x + 0x28
7. p_bit_timer (static)	long unsigned				x + 0x30
8. reject_timer (static)	long unsigned				x + 0x38
9. busy_state_timer (static)	long unsigned				x + 0x40
Specific methods	***m/o***				

Attribute description

logical_name

Identifies the "ISO/IEC 8802-2 LLC Type 2 setup" object instance.

transmit_window_size_k

The transmit window size (k) shall be a data link connection parameter that can never exceed 127. It shall denote the maximum number of sequentially numbered I PDUs that the sending LLC may have outstanding (i.e., unacknowledged). The value of k is the maximum number by which the sending LLC send state variable V(S) can exceed the N(R) of the last received I PDU.

See sub-clause 7.8.4 *Transmit window size, k.*

receive_window_size_rw

The receive window size (RW) shall be a data link connection parameter that can never exceed 127. It shall denote the maximum number of unacknowledged sequentially numbered I PDUs that the local LLC allows the remote LLC to have understanding. It is transmitted in the information field of XID (see ISO/IEC 8802-2 5.4.1.1.2) and applies to the XID sender. The XID receiver shall set its transmit window (k) to a value less than or equal to the receive window of the XID sender to avoid overrunning the XID sender.

See ISO/IEC 8802-2 sub-clause 7.8.6 *Receive window size, RW.*

max_octets_i_pdu_n1

N1 is a data link connection parameter that denotes the maximum number of octets in an I PDU. Refer to the various MAC descriptions to determine the precise value of N1 for a given medium access method. LLC itself places no restrictions on the value of N1. However, in the interest of having a value of N1 that all users of Type 2 LLC may depend upon, all MACs must at least be capable of accommodating I PDUs with information fields up to an including 128 octets in length.

See ISO/IEC 8802-2 sub-clause 7.8.3 *Maximum number of octets in an I PDU, N1.*

max_number_transmissions_n2

N2 is a data link connection parameter that indicates the maximum number of times that a PDU is sent following the running out of the acknowledgment timer, the P-bit timer, the reject timer, or the busy-state timer.

See ISO/IEC 8802-2 sub-clause 7.8.2 *Maximum number of transmissions, N2.*

acknowledgement_timer

The acknowledgment timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive an acknowledgment to one or more outstanding I PDUs or an expected response PDU to a sent unnumbered command PDU.

The unit is seconds.

See ISO/IEC 8802-2 sub-clause 7.8.1.1 *Acknowledgement timer.*

p_bit_timer

The P-bit timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a PDU with the F bit set to “1” in response to a sent Type 2 command with the P bit set to “1”.

The unit is seconds.

See ISO/IEC 8802-2 sub-clause 7.8.1.2 *P-bit timer.*

reject_timer

The reject timer is a data link connection parameter that shall define the time interval during which the LLC shall expect to receive a reply to a sent REJ PDU.

The unit is seconds.

See ISO/IEC 8802-2 sub-clause 7.8.1.3 *Reject timer.*

busy_state_timer

The busy-state timer is a data link connection parameter that shall define the timer interval during which the LLC shall wait for an indication of the clearance of a busy condition at the other LLC.

The unit is seconds.

See ISO/IEC 8802-2 sub-clause *Busy-state timer.*

B.9 ISO/IEC 8802-2 LLC Type 3 setup (class_id: 59)

An instance of the ISO/IEC 8802-2 LLC Type 3 setup class holds the parameters necessary to set up the ISO/IEC 8802-2 LLC layer in Type 3 operation.

ISO/IEC 8802-2 LLC Type 3 Setup	**0...n**	**class_id = 59, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				X
2. max_octets_acn_pdu_n3 (static)	Long unsigned				x + 0x08
3. max_number_transmissions_n4 (static)	unsigned				x + 0x10
4. acknowledgement_time_t1 (static)	Long unsigned				x + 0x18
5. receive_lifetime_var_t2 (static)	Long unsigned				x + 0x20
6. transmit_lifetime_var_t3 (static)	Long unsigned				x + 0x28
Specific methods	***m/o***				

Attribute description

logical_name

Identifies the “ISO/IEC 8802-2 LLC Type 3 setup” object instance

max_octets_acn_pdu_n3

N3 is a logical link parameter that denotes the maximum number of octets in an ACn command PDU. Refer to the various MAC descriptions to determine the precise value of N3 for a given medium access method. LLC places no restrictions on the value of N3.

See ISO/IEC 8802-2 sub-clause 8.6.2 *Maximum number of octets in an ACn command PDU, N3.*

max_number_transmissions_n4

N4 is a logical link parameter that indicates the maximum number of times that an ACn command PDU is sent by LLC trying to accomplish a successful information exchange. Normally, N4 is set large enough to overcome the loss of a PDU due to link error conditions. If the medium access control sublayer has its own retransmission capability, the value of N4 may be set to one so that LLC does not itself requeue a PDU to the medium access control sublayer.

See ISO/IEC 8802-2 sub-clause 8.6.1 *Maximum number of transmissions, N4.*

acknowledgement_time_t1	The acknowledgment time is a logical link parameter that determines the period of the acknowledgment timers, and as such shall define the time interval during which the LLC shall expect to receive an ACn response PDU from a specific LLC from which the LLC is awaiting a response PDU. The acknowledgment time shall take into account any delay introduced by the MAC sublayer and whether the timer is started at the beginning or at the end of the sending of the ACn command PDU by the LLC. The proper operation of the procedure shall require that the acknowledgment time be greater than the normal time between the sending of an ACn command PDU and the reception of the corresponding ACn response PDU. If the medium access control sublayer performs its own retransmissions and if the logical link parameter N4 is set to one to prevent LLC from re-queuing a PDU, then the acknowledgment time T1 may be set to infinity, making the acknowledgment timers unnecessary. The unit is seconds. Infinity is indicated by all bits set to 1. See ISO/IEC 8802-2 sub-clause 8.6.4 *Acknowledgement time, T1.*
receive_lifetime_var_t2	This time value is a logical link parameter that determines the period of all of the receive variable lifetime timers. T2 shall be longer by a margin of safety than the longest possible period during which the first transmission and all retries of a single PDU may occur. The margin of safety shall take into account anything affecting LLCs perception of the arrival time of PDUs, such as LLC response time, timer resolution, and variations in the time required for the medium access control sublayer to pass received PDUs to LLC. If the destruction of the received state variables is not desired, the value of time T2 may be set to infinity. In this case the receive variable lifetime timer need not be implemented. The unit is seconds. Infinity is indicated by all bits set to 1. See ISO/IEC 8802-2 sub-clause 8.6.5 *Receive lifetime variable, T2.*
transmit_lifetime_var_t3	This time value is a logical link parameter that determines the minimum lifetime of the transmit sequence state variables. T3 must be longer by a margin of safety than 1) the logical link variable T2 at stations to which ACn commands are sent; and 2) the longest possible lifetime of an ACn command-response pair. The lifetime of an ACn command-response pair must take into account the sum of processing time, queuing delays, and transmission time for the command and response PDUs at the local and remote stations. If the destruction of the transmit state variables is not desired, the value of time T3 may be set to infinity. Note, if the receive variable lifetime parameter, T2 is set to infinity at remote stations to which ACn commands are sent, then the T3 parameter must be set to infinity at the local station. The unit is seconds. Infinity is indicated by all bits set to 1. See ISO/IEC 8802-2 sub-clause 8.6.6 *Transmit lifetime variable, T3.*

APPENDIX C: SPECIFICATION OF IMAGE TRANSFER MECHANISM

C.1 Image Transfer Interface Class (class_id:18)

This interface class allows transfer of Firmware Image(s) to COSEM Servers. Image transfer is performed with several steps. First Image transfer is enabled and initiated. Then Image is transferred with blocks. Block transfer can be performed individually or with broadcast service. When completed transfer continues with transfer of missing blocks and finally with verification and activation of the Image.

Image Transfer		**0..n**	**class_id = 18, version = 0**			
Attribute(s)		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. image_block_size	(static)	octet-string				x + 0x08
3. image_block_transfer	(dynamic)	image_block_transfer _type				x + 0x10
4. image_missing_blocks	(dynamic)	bit-string				x + 0x18
5. image_first_missing_block_offset	(dynamic)	double-long-unsigned				x + 0x20
6. transfer_enabled	(static)	boolean				x + 0x28
7. images_info	(dynamic)	array				x + 0x30
Specific methods (if required)		***m/o***				
1. init_transfer		O				x + 0x38
2. verify_image		O				x + 0x40
3. verify_and _activate_image		O				x + 0x48

Attribute description

logical_name	Identifies the “Image Transfer” object instance. For logical names, see chapter 5.13
image_block_size	is size of Image Block expressed in octets. Image is divided into consecutive Image Blocks of data.

image_block_transfer	Provides the image_block_transfer for image transfer . Image blocks are delivered
	image_block_transfer_type ::= structure
	{
	image_block_offset double-long-unsigned,
	image_block_value octet-string
	}

image_missing_blocks	Provides information about missing blocks. Each bit in bit-string provides information about individual Image Block.

	0 = Missing 1 = Transferred
image_first_missing_block_offset	Provides offset of first missing blocks.
image_transfer_enabled	is enabled status of image transfer.

images_info	Provides the image_info for all images.
	array image_info
	image_info ::= structure
	{
	image_size double-long-unsigned,
	image_identification octet-string,
	image_signature octet-string
	}
	image_size is size of the image expressed in octets
	image_identification is identification of the image and can contain information like manufacturer, device type, version information, etc.
	image_signature is signature of the image which is calculated from the image data. Algorithms applied for calculation of signature can be MD5, SHA-1 or other.

Method description

init_transfer(data)	Initialize image transfer data double-long-unsigned () data contains size of the image to be transferred in octets.
image_verify(data)	Verifies the integrity of the image before activation data long-unsigned (0)
image_verify_and_activate(data)	Verifies the integrity of the image before activation and activates the image. data long-unsigned (0)