P3 Companion Standard

Dutch Smart Meter Requirements

Date:	**March 14th, 2014**
Version:	**4.0.7**
Status:	**Final**

CONTENTS

1 INTRODUCTION

1.1 Scope

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

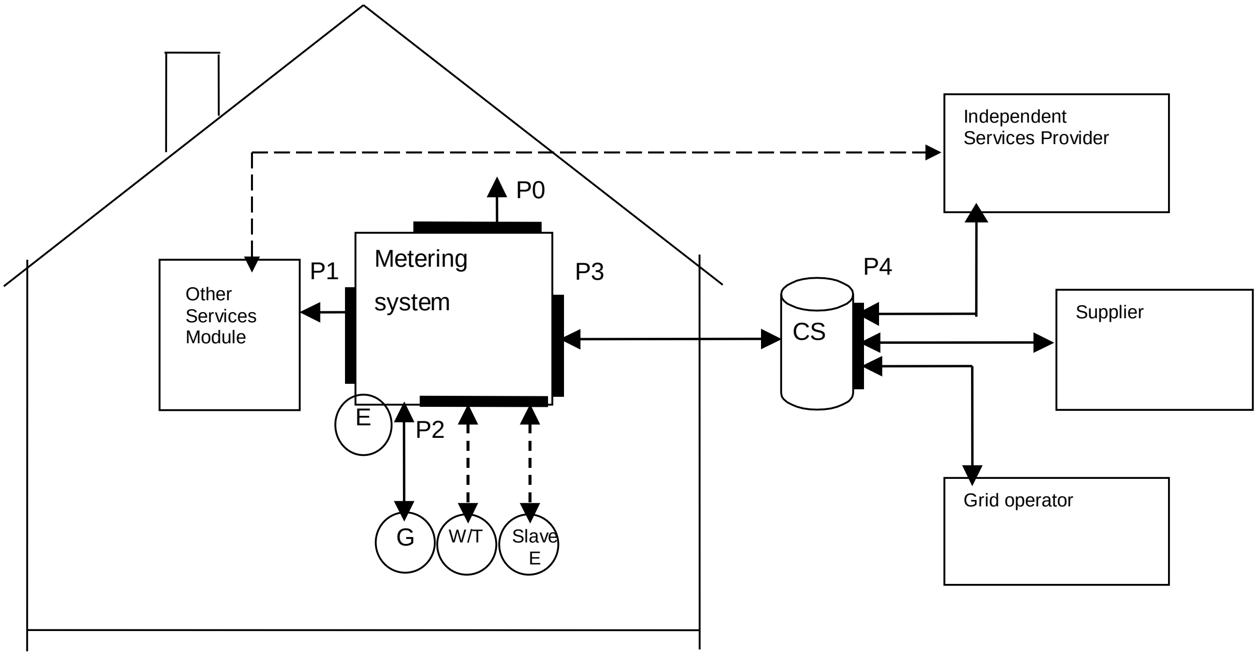


Figure 1.1: Meter interfaces overview

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 manufacturers 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. The P0 and the P3 port provide access to the same OBIS objects with the same authorisation settings. Authentication on P3 and P0 might differ depending on the configuration settings described elsewhere in this document. The 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.

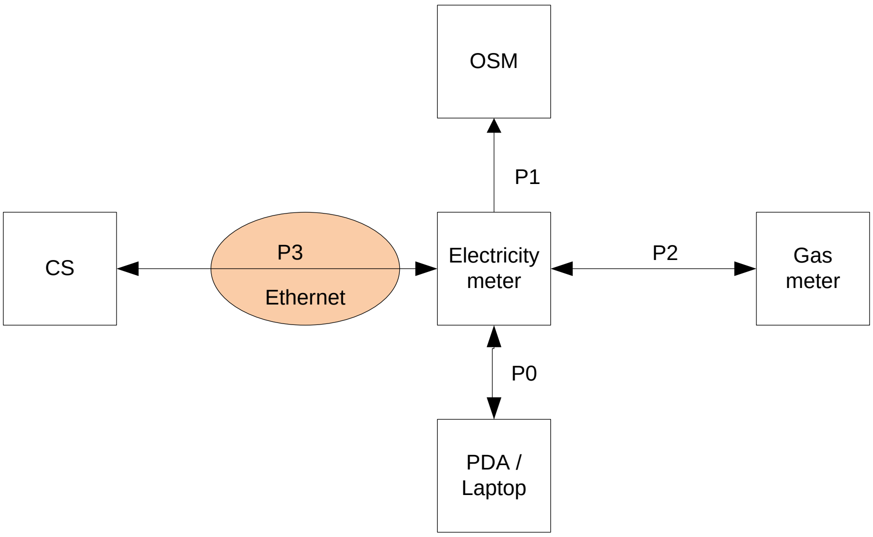


Figure 1.2: DLMS/COSEM infrastructure P3

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.10, 2009	Blue book, COSEM Identification System and Interface Classes
2	DLMS UA 1000-2 ed.7, 2007	Green book, DLMS/COSEM Architecture and Protocols
3	DLMS UA Green Book 7.0, Amendment 2	Green book, DLMS/COSEM Architecture and Protocols, Amendment 2
4	DLMS UA 1001-1 ed.3, 2007	Yellow book, DLMS/COSEM Conformance Testing Process
5	DLMS UA 1002: ed.1, 2003	White book, COSEM Glossary of Terms
6	IEC 62056-21 Ed 1.0:2002	Electricity metering – Data exchange for meter reading, tariff and load control – Part 21: Direct local data exchange
7	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
8	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

Ref No	Document	Description
9	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
10	IEC 62056-53 Ed 2.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 53: COSEM Application layer
11	IEC 62056-61 Ed 2.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 61: OBIS Object identification system
12	IEC 62056-62 Ed 2.0:2006	Electricity metering – Data exchange for meter reading, tariff and load control – Part 62: Interface classes
13	NTA 8130 NL:2007	Netherlands Technical Agreement -"Minimum set of functions for metering of electricity, gas and thermal energy for domestic customers"
14	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
15	ISO/IEC 13239	Information technology -- Telecommunications and information exchange between systems -- High-level data link control (HDLC) procedures
16	EN 13757-2	Communication systems for and remote reading of meters – Part 2: Physical and link layer
17	EN 13757-3 : 2004	Communication systems for and remote reading of meters – Part 3: Dedicated application layer
18	P&S guidelines 1.5	Guidelines from the Privacy & Security work group version 1.5
19	AmvB	Algemene maatregel van Bestuur "Besluit op afstand uitleesbare meet-inrichtingen"
20	ISO/IEC 8825 Ed. 3:2002	Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)
21	IEC 61334-6 Ed 1.0:2000	Distribution automation using distribution line carrier systems – Part 6: A-XDR encoding rule

Table 1-1: Normative References

Remark:

1. If there are functionalities mentioned in the Blue book, necessary for the implementation, which are not present in this P3 document then the blue book will be leading for the implementation. Furthermore there shall be consensus between the different parties (meter vendors) on how to implement these functionalities, in order to maintain interoperability.

1.4 Document list

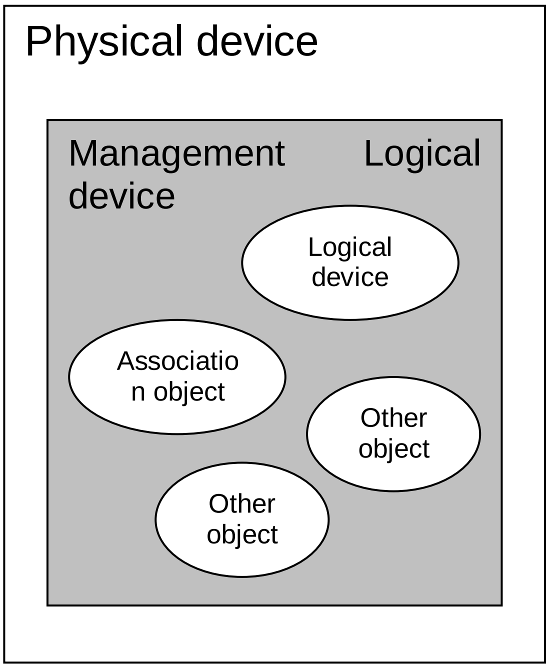
Following table shows the complete set of documents that build up the Dutch Smart Meter Requirements, of which this document is a part of.

#	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]	P1	Companion standard P1
[3]	P2	Companion standard P2
[4]	P3	Companion standard P3
[5]	GPRS	Additional document describing the requirements for the GPRS infrastructure as part of the Dutch Smart Meter Specification.

Table 1-2: Document List

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 can have 3 associations: Public client (client Id 16), Management client (client Id 1) and Pre-established client (client Id 102). The access rights of these clients are indicated in the "P", "M", and "Pr" columns used in the object descriptions of this document.

There will be no direct access to the M-Bus device (connected to the E meter via P2) via the P3 interface. Access to the M-Bus device is provided through 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 be allowed to read metering data, or perform any programming.

Therefore the services within the public client are restricted to:

- Block transfer with Get
- Get
- Selective access

2.1.2 *Pre established client*

The pre-established client is for broadcast purposes. It could be used when connected via Ethernet.

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

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

In this version of the DSMR, no broadcast functionality is offered. (No objects that can be accessed through this client.)

2.1.3 *Management client*

The management client (Client Id 1) is the client to be used by the central system in case of GPRS meters for regular point to point connections with the meters.

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

- Block transfer with Get
- Block transfer with Set
- Set
- Get
- Set request with list
- Get request with list
- Selective access
- Action

The maximum number of simultaneous requests is dependent on the server_max_receive_pdu_size, which is set to 1024.

The maximum number of multiple references for Set request with list and Get request with list is 16.

2.2 Access right

For each of the attributes of all the objects defined in this document, the access rights for each of the clients are defined by defining Get and Set access (indicated with R and W) for attributes and action access (indicated with X). No access is indicated with white spaces.

If a client tries to access an attribute or method for which appropriate access is not granted, the server will return a read-write-denied result. This is not considered a protocol error..

3 COMMUNICATION PROFILES AND SERVICES, SECURITY

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

There is one communication profile selected for the Dutch Smart Meter.
These are:

- GPRS

DLMS/COSEM will only use the 'pull' mechanism for the application layer[1].

How the E-meter will establish a GPRS connection is described in the profiles part (see chapter 3.1.1).

The profile for GPRS meters uses standard COSEM TCP/IP profile as defined in IEC 62056-47 or in DLMS UA Green Book edition 7. The TCP/UDP based profile is given in figure 3.1.

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

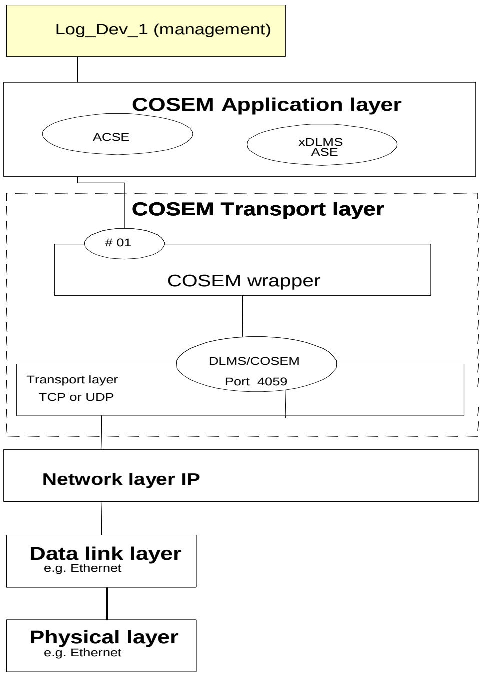


Figure 3.1: TCP/UDP based profile

[1] The operation principle used in this document is based on client / server model. For all services the client always sends a request and the server sends back the response to the this request.

The required setup objects for TCP/UDP, IPv4 and PPP are described below. For the detailed attribute and method descriptions see chapter 4 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**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:25.0.0.255**		R	
2	TCP-UDP_port	long-unsigned	Default 4059		R	
3	IP_reference	octet-string			RW	
4	MSS	long-unsigned	Min=40, max=65535, default=576		RW	
5	nb_of_sim_conn	unsigned	Value=1[2]		R	
6	inactivity_time_out	long-unsigned	Default=300		RW	
	Specific methods	***m/o***				

Table 3-1: TCP-UDP Setup

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 PPP), then an instance of the TCP-UDP setup object is necessary for each of them.

NOTE: The value “b” in the OBIS code always means channel. Since a given meter can only have one communication profile the values of “b” shall be “0” for all communication profiles

[2] This value describes the number of simultaneous connections for the COSEM protocol. An additional TCP connection is optional for setting up the connection using push mechanisms as described in section 9.

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.**				**P**	**M**	**Pr**
1	Logical name	octet-string	**0-0:25.1.0.255**		R	
2	DL_reference port	octet-string			RW	
3	IP_address	double-long-unsigned			RW	
4	multicast_IP_address	array			RW	
5	IP-options	array			RW	
6	Subnet_mask	double-long-unsigned			RW	
7	gateway_IP_address	double-long-unsigned			RW	
8	use_DHCP_flag	boolean			RW	
9	primary_DNS_address	double-long-unsigned			RW	
10	secondary_DNS_address	double-long-unsigned			RW	
	Specific methods	***m/o***				
1	add_mc_IP_address (data)	o				
2	delete_mc_IP_address (data)	o				
3	get_nbof_mc_IP_addresses (data)	o				

Table 3-2: IPv4 Setup

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 interface 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.**				**P**	**M**	**Pr**
1	logical_name	Octet-string	**0-0:25.3.0.255**		R	
2	PHY_reference	Octet string			RW	
3	LCP_options	LCP_options_type			RW	
4	IPCP_options	IPCP_options_type			RW	
5	PPP_authentication	PPP_auth_type			RW	
	Specific methods	***m/o***				

Table 3-3: PPP Setup

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 two specific communication profiles are described in the following paragraphs. TCP is used (and not UDP) for both DLMS/COSEM protocol and also for the wake-up message as defined in section 8.

3.1.1 *GPRS communication profile*

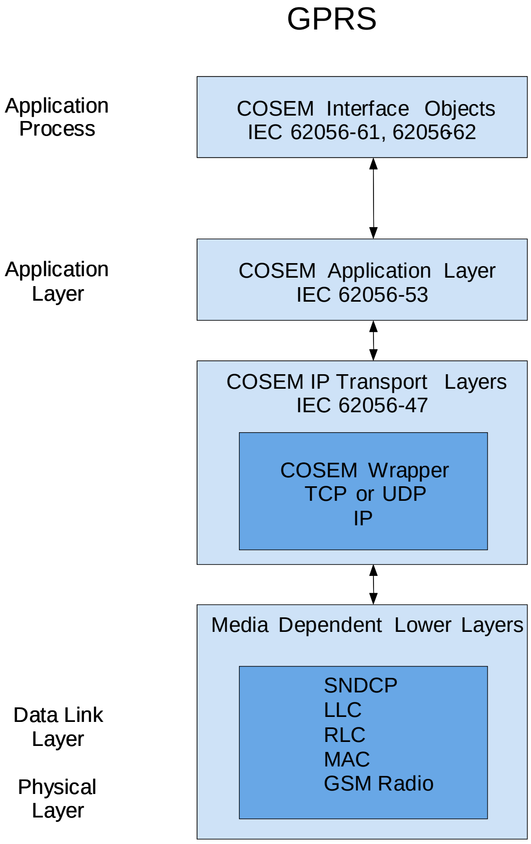


Figure 3.2: GPRS communication profile

A “GPRS modem setup” object stores all the necessary data for GPRS modem management. For the detailed attribute and method descriptions see chapter 4 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.**				**P**	**M**	**Pr**
1	logical_name	octet-string	**0-0:25.4.0.255**		R	
2	APN	octet-string			RW	
3	PIN_code	long-unsigned			RW	
4	quality_of_service	structure			RW	
	Specific methods	***m/o***				

Table 3-4: GPRS Modem Setup

The connection setup mechanism used by the grid operators as well as the specific GPRS requirements are described in the separate DSMR GPRS requirements document.

Note: The network access information shall not be saved on the SIM/USIM cards or Chip/Soft SIM cards.

3.2 COSEM Security

3.2.1 *General*

After commissioning, the *security policy* attribute in the Security Setup object (OBIS 0-0:43.0.0.255) shall have the value 3 (Message Encryption and Message Authentication). If this value is not the factory setting, the value can only be changed via the method 'security activate'. Once this attribute has the value 3, it is not possible to lower the value.

3.2.2 *Access Security*

The following table defines the requirements for Data Access Security for the Public and Management clients via the P0 and P3:

Data security Acces security	no security	authenticated	encrypted	authenticated and encrypted	Remark about Data security (in DLMS "security policy): See Blue Book, page 71: NOTE The security policy can only be strengthened. Strengthening is setting the policy to a higher numeric value.
No security (Lowest Level Security)	allowed for Public Client only	not allowed	not allowed	not allowed	
LLS (Low Level Security)	not allowed	not allowed	not allowed	not allowed	
HLS2	not allowed	not allowed	not allowed	not allowed	
HLS3	allowed	allowed	allowed	allowed	See DSMR M 4.4.8a and 4.4.8b
HLS4	allowed	allowed	allowed	allowed	See DSMR M 4.4.8a and 4.4.8b
HLS5	allowed	allowed	allowed	allowed	See DSMR M 4.4.8a and 4.4.8b

HLS 3 and 4 do use the HLS secret
HLS 5 does use Authentication Key and Encryption Key
The same encryption methods (data security) must be used for HLS 5, 4 en 3

Access via P0 is allowed for commissioning purposes and maintenance purposes. The access can be toggled via the *P0_enable* bit.

3.2.3 *Transport Security*

3.2.3.1 *Data exchange*

For data transport security for data exchange, the following applies:

If the Public Client has established an Association with the Management Logical Device, data exchange may be done in an unciphered application context. Neither message encryption nor message authentication is required.

If the Management Client has established an Association with the Management Logical Device, the following requirements apply:

1. All data exchange shall take place in a ciphered application context, indicated by *context_id* 3 (Logical Name Services and Ciphering) for the field *application_context_name* in the AARQ.

2. Only 'Message Encryption & Message Authentication' with the mechanisms provided by DLMS/COSEM Security Suite 0 shall be used. This security suite contains the following Authentication and Encryption Algorithms:

Security Suite ID	Authentication Algorithm	Encryption Algorithm
0	AES-GCM-128	AES-GCM-128 and AES-128 for key wrapping

   Using both Message Authentication and Message Encryptions means that the *security policy* attribute in the Security Setup object (Logical Name 0-0:43.0.0.255) shall have the value 3.

3. Resulting from 2, the *InitiateRequest* and *InitiateResponse* xDLMS APDU's in the AARQ and RLRQ respectively shall be encrypted and authenticated.

4. All other fields in the AARQ and RLRQ shall be transmitted in clear text, according to the examples in the Green Book.

5. Only Global Encryption Keys shall be used for message encryption. The use of dedicated keys that are transported via the AARQ is NOT allowed.

If the Management Client has established an Association with the Management Logical Device, data exchange in an unciphered application context (with *security policy* set to 0), is intended to be used for the purposes of commissioning the meter via P0.

Transfer Key handling

The E-meters shall interpret the octet string (containing the wrapped key) inside the data field of the transfer_key method of the M-Bus client object as described in DLMS blue book version 10 as MSB (most significant byte) first. The E-meters shall internally convert this octet string to make it LSB (least significant byte) first so that the converted octet string is in the appropriate format to be transferred over P2 as described in DSMR P2 companion standard paragraph 4.2.

High Level Security (HLS)

HLS provides authentication of the client and server when establishing an association. The authentication mechanism “mechanism_id(5)” as described in the DLMS/COSEM Green Book shall be used for entity authentication.

Frame Counters

To detect and counter replay attacks the reuse of frame counters must be detected. Frame counters are also used in the Initialization Vector of the encryption algorithm. There is a Frame Counter for communication between CS and Meter and a Frame Counter for communication between Meter and CS.

Requirements for Frame Counter used for communication between CS and Meter:

- Within an Application Association, frame counters must be incremented for each frame send by the CS.
- The E-meter must validate that each subsequent frame has a frame counter that is exactly 1 higher than the frame counter of the previous message
- The E-Meter must release the Application Association immediately when messages are received with frame counters that are not in sequence.
- The E-Meter shall not send an error when receiving a frame counter that is not in sequence.
- The frame counter for the first frame in any session is determined by the CS.
- The E-meter shall use the first received Frame Counter within an Application Association as starting point to validate the Frame Counters of each subsequent message within that Application Association.
- The E-Meter shall not validate the first Frame Counter used in any session; the CS is responsible to keep the Frame Counter unique within the lifetime of a global encryption key to match the DLMS requirements regarding unique values for the Initialization Vector. Note that the first Frame Counter in any session is part of the HLS challenge response authentication mechanism which makes validating the first Frame Counter unnecessary.

Requirements for Frame Counter used for communication between Meter and CS:

- The meter shall not have a mechanism which creates a relation between the two frame counters. The frame counters for messages from meter to CS and the messages from CS to meter are independent from each other.
- The frame counter used in the messages send by the meter should be incremented by exactly one for each frame send.
- The CS will validate the Frame Counter and check that the received Frame Counter is higher than in a previous message.

- When a CS receives a message with a Frame Counter which is out of sequence it will not send an error to the E-Meter
- The E-Meter resets the Frame Counter to zero when the Frame Counter reaches the maximum value. Note that it is the responsibility of the CS that the global encryption key has been changed before the Frame Counter reaches a value that has been used with same global encryption key. The Frame Counters (used for communication between CS->Meter and Meter->CS) may be reset to zero after a successful change of the Global Encryption Key.

Encryption of Initiate Request

The initiate request (part of the application association) must be encrypted as defined in The Green Book update as shown in Annex C.

3.3 Data efficiency using TCP/IP

In order to minimize datacom costs, data sessions need to be as short as possible and the amount of data transferred in that time needs to be maximized.

- Session duration need to be as short as possible (Ideally < 30 seconds)
- Packet size needs to be optimized (Ideally 500 bytes)
- The amount of unnecessary signaling packets (for instance empty packets or empty ACKs) needs to be minimized (Ideally greater than 80 % Data to 20 % signaling bytes).

3.4 Coding principles

The DLMS COSEM protocol use standardized coding rules for data types as mentioned in the Green Book [2]. Two standards are used:

- ACSE APDUs (used in Application Associations) are coded in BER (ISO/IEC 8825) [21];
- xDLMS APDUs are coded in A-XDR [22];

This section gives some guidelines and explanation on the coding of data types in the xDLMS APDUs.

This companion standard uses the (basic) data types in the class attributes and method specification as listed in Table 3-5. The complex data types defined in Green Book [2] (e.g. LCP_options_type) are not listed; they are built from the basic types from Table 3-5..

Data Type	**Coding rule**
unsigned	Fixed length unsigned (8 bits) integer; section 6.1.1.1[22]
long-unsigned	Fixed length unsigned (16 bits) integer; section 6.1.1.1[22]
double-long-unsigned	Fixed length unsigned (32 bits) integer; section 6.1.1.1[22]
integer	Fixed length signed (8 bits) integer; section 6.1.1.2[22]
enum	Fixed length signed (8 bits) integer; section 6.3[22]
boolean	One byte with value zero (0) or non-zero; section 6.2[22]
octet-string	When the length is not clear from the context then a variable length byte string shall be used; section 6.5.2[22] When the length is clear from the context then a fixed length byte string shall be used; section 6.5.1[22]

	An empty octet string shall be encoded as H’09 (tag octet-string) and H’00 length so a code of 2 bytes
octet-string[Length]	Fixed length byte string; section 6.5.1[22]
array	When the length is not clear from the context then a variable length SEQUENCE OF values shall be used; section 6.10.2[22] When the length is clear from the context then a fixed length SEQUENCE OF values shall be used; section 6.10.1[22] An array[0] shall be encoded as H’01 (tag array) and H’00 (length) so a code of 2 bytes.
array[length]	Fixed length SEQUENCE OF values shall be used; section 6.10.1[22] In case of an array with script tables null-data (tag = 0) should be used for not used entries.

Table 3-5: Coding rules of used basic data types.

3.5 Exception handling

The COSEM server shall comply to the described exception handling in Green Book[2]. In addition the following exception rules are applicable:

- A GetRequest-WithList shall be executed as far as possible and the response shall contain a list of all available information and an error for the missing objects.
- In cases where an action within the E meter cannot be performed in the same communication session with the Central System, the E meter shall return “success” to the Central System as soon as the command was received successfully by the E meter. If the action is executed in the same communication session the feedback depends on the actual result of that action.

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 Meters are described in more detail, in the mentioned chapters an overview will be given of all required profiles. Furthermore an explanation of the event and error handling is described in paragraph 4.2 of this chapter.

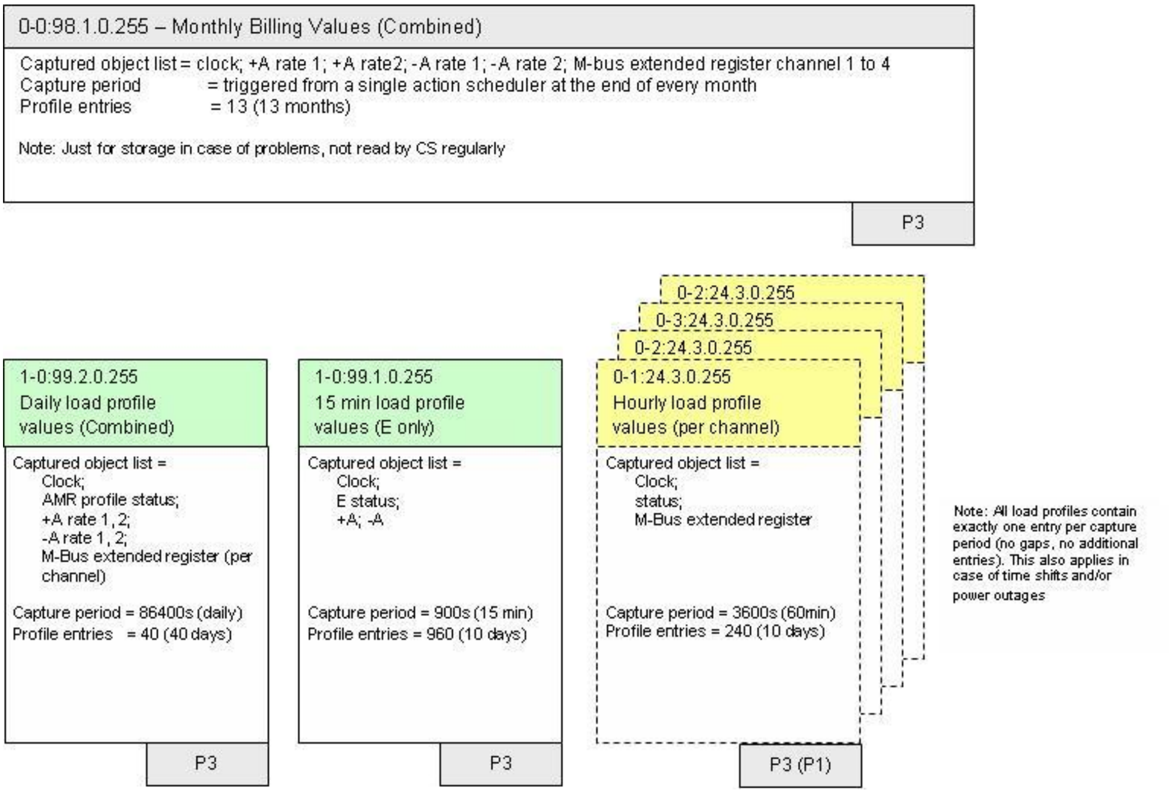
4.1 Profile structure Dutch Smart Meter

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 (combined)*" and "*15 min load profile values (E Only)*" 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 rate2; -A rate 1; -A rate 2; M-bus extended register channel 1 to 4
Capture period = triggered from a single action scheduler at the end of every month
Profile entries = 13 (13 months)

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

P3

1-0:99.2.0.255
Daily load profile values (Combined)

Captured object list =
Clock;
AMR profile status;
+A rate 1, 2;
-A rate 1, 2;
M-Bus extended register (per channel)

Capture period = 86400s (daily)
Profile entries = 40 (40 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 = 900s (15 min)
Profile entries = 960 (10 days)

P3

0-2:24.3.0.255
0-3:24.3.0.255
0-2:24.3.0.255
0-1:24.3.0.255
Hourly load profile values (per channel)

Captured object list =
Clock;
status;
M-Bus extended register

Capture period = 3600s (60min)
Profile entries = 240 (10 days)

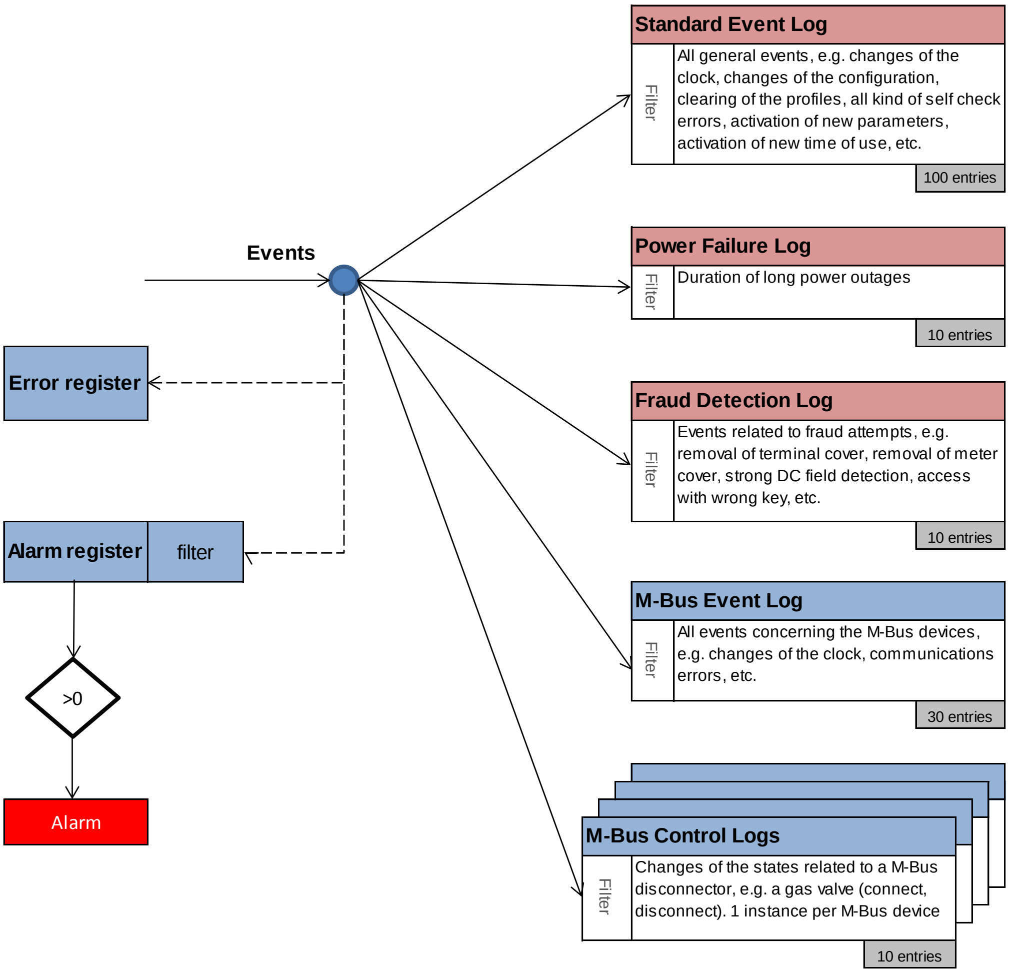
P3 (P1)

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 [13].


Events
Error register
Alarm register
filter
>0
Alarm
Standard Event Log
Filter
All general events, e.g. changes of the clock, changes of the configuration, clearing of the profiles, all kind of self check errors, activation of new parameters, activation of new time of use, etc.
100 entries
Power Failure Log
Filter
Duration of long power outages
10 entries
Fraud Detection Log
Filter
Events related to fraud attempts, e.g. removal of terminal cover, removal of meter cover, strong DC field detection, access with wrong key, etc.
10 entries
M-Bus Event Log
Filter
All events concerning the M-Bus devices, e.g. changes of the clock, communications errors, 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. 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 that triggered it. Every event is assigned to one event log (event filter) and it is only stored there. This assignment is not configurable.

Event codes:

Number	Name	Description	Standard Event log	Fraud Detection 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
1	Power Down	Indicates a complete power down of the device in any of the phases. Please note that this is related to the device and not necessarily to the network.	X						
2	Power Up	Indicates that the device is completely powered again (in case of a polyphase meter; all phases) after a complete power down in any of the phases.	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	tariff shift times (TOU) activated	Indicates that the passive tariff shift times (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						
17	Firmware ready for activation	Indicates that the new firmware has been successfully downloaded and verified, i.e. it is ready for activation	X						

Number	Name	Description	Standard Event log	Fraud Detection 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
18	Firmware activated	Indicates that a new firmware has been activated	X						
19	Tariff Shift Time (TOU)	Change of tariff shift times has occurred	X						
20	Succesfull selfcheck after Firmwareupdate	Indicates that the first selfcheck after a firmwareupdate was performed succesfully.	X						
21-39	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	Failed login attempt	Indicates that a user tried to gain access with wrong authentication credentials (intrusion detection)		X					
47	Configuration change	Indicates that configuration is activated or de-activated after the meter was installed.		X					
48-59	reserved for future use								
60	Reserved for backwards compatibility								
61	Reserved for backwards compatibility								
62	Reserved for backwards compatibility								
63	Reserved for backwards compatibility								
64	Reserved for backwards compatibility								
65	Reserved for backwards compatibility								
66	Reserved for backwards compatibility								
67	Reserved for backwards compatibility								
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				

Number	Name	Description	Standard Event log	Fraud Detection 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
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				
104	Clock adjusted M-Bus channel 1	Indicates that the clock has been adjusted. (Only in case of a large time adjustment)			X				
105	New M-Bus device discovered channel 1	Indicates that a new M-Bus Device has been detected on channel 1 of the M-Bus			X				
106	Permanent error from M-Bus device channel 1	Indicates that a Permanent error has been received from the M-Bus device. The Permanent error can be a self-check error, or any other fatal device error that requires a service action			X				
107-109	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. (Only in case of a large time adjustment)			X				
115	New M-Bus device discovered channel 2	Indicates that a new M-Bus Device has been detected on channel 2 of the M-Bus			X				
116	Permanent error from M-Bus device channel 2	Indicates that a Permanent error has been received from the M-Bus device. The Permanent error can be a self-check error, or any other fatal device error that requires a service action			X				
117-119	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. (Only in case of a large time adjustment)			X				

Number	Name	Description	Standard Event log	Fraud Detection 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
125	New M-Bus device discovered channel 3	Indicates that a new M-Bus Device has been detected on channel 3 of the M-Bus			X				
126	Permanent error from M-Bus device channel 3	Indicates that a Permanent error has been received from the M-Bus device. The Permanent error can be a self-check error, or any other fatal device error that requires a service action			X				
127-129	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. (Only in case of a large time adjustment)			X				
135	New M-Bus device discovered channel 4	Indicates that a new M-Bus Device has been detected on channel 4 of the M-Bus			X				
136	Permanent error from M-Bus device channel 4	Indicates that a Permanent error has been received from the M-Bus device. The Permanent error can be a self-check error, or any other fatal device error that requires a service action			X				
137-159	reserved for future use								
160	Reserved for backwards compatibility								
161	Reserved for backwards compatibility								
162	Reserved for backwards compatibility								
163	Reserved for backwards compatibility								
164	Valve alarm M-Bus channel 1	Indicates that a valve alarm has been registered.				X			
165-169	reserved for future use								
170	Reserved for backwards compatibility								
171	Reserved for backwards compatibility								

Number	Name	Description	Standard Event log	Fraud Detection 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
172	Reserved for backwards compatibility								
173	Reserved for backwards compatibility								
174	Valve alarm M-Bus channel 2	Indicates that a valve alarm has been registered.					X		
175-179	reserved for future use								
180	Reserved for backwards compatibility								
181	Reserved for backwards compatibility								
182	Reserved for backwards compatibility								
183	Reserved for backwards compatibility								
184	Valve alarm M-Bus channel 3	Indicates that a valve alarm has been registered.						X	
185-189	reserved for future use								
190	Reserved for backwards compatibility								
191	Reserved for backwards compatibility								
192	Reserved for backwards compatibility								
193	Reserved for backwards compatibility								
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 and is of type Unsigned. To identify the event code in the different event logs, the following OBIS codes are used in the capture_objects attribute of the event log:

0-0:96.11.0.255	Standard event Log
0-0:96.11.1.255	Fraud detection Log
0-0:96.11.3.255	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 3 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. 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

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

Contains information related to long power outages, i.e. duration and end time of a long power outage.
Structure: Timestamp – duration of long power failures
Minimum size: 10 entries

Fraud Detection Event 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 magnetic field detection, access with wrong keys, 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 event related to an M-Bus valve error.
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 at anytime.

Depending on the type of error, some errors clear themselves if the reason for the error has disappeared. Others must be cleared via the management client. Clearing is possible by the Management Client (object 0-0:97.97.0.255) for errors and object 0-0:97.98.0.255 for alarms). Nevertheless the events are stored in one of the event logs.All logical errors as mentioned in the DSMR, as a response on a command, are mapped on the DLMS error "other reason".

Normal Error Codes

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

Group	Byte	Bit	Meaning	Events
Other Errors	1 (LSB)	0	Clock invalid	6
		1	Replace battery	7
		2	Power Up	2
		3	not used	
		4	not used	
		5	not used	
		6	not used	
		7	not used	
Critical Errors	2	0	Program memory error	12
		1	RAM Error	13
		2	NV memory Error	14
		3	Measurement System Error	16
		4	Watchdog error	15
		5	Fraud attempt	40, 42, 44, 46, 47
		6	not used	
		7	not used	
M-Bus Errors	3	0	Communication error M-Bus channel 1	100
		1	Communication error M-Bus channel 2	110
		2	Communication error M-Bus channel 3	120
		3	Communication error M-Bus channel 4	130
		4	Fraud attempt M-Bus channel 1	103
		5	Fraud attempt M-Bus channel 2	113
		6	Fraud attempt M-Bus channel 3	123
		7	Fraud attempt M-Bus channel 4	133
Reserved	4 (MSB)	0	New M-Bus device discovered channel 1	105
		1	New M-Bus device discovered channel 2	115
		2	New M-Bus device discovered channel 3	125
		3	New M-Bus device discovered channel 4	135
		4	not used	
		5	not used	
		6	not used	
		7	not used	

For a detailed description see the corresponding event. Critical errors, New M-Bus device discovered channel x, Power Up and the replacement of the battery must be cleared via the management client, all others clear themselves if the corresponding error condition has disappeared.

4.2.4 *Alarm Handling*

A selection of events can be made which can be treated as alarms (alarm filter). If one of these selected events occurs, the corresponding flag in the alarm register is set and an alarm is then raised 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 the management client (acknowledgement).

Typically 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.

Group	Byte	Bit	Meaning	Events
Other Alarms	1 (LSB)	0	Clock invalid	6
		1	Replace battery	7
		2	Power Up	2
		3	not used	
		4	not used	
		5	not used	
		6	not used	
		7	not used	
Critical Alarms	2	0	Program memory error	12
		1	RAM Error	13
		2	NV memory Error	14
		3	Measurement System Error	16
		4	Watchdog error	15
		5	Fraud attempt	40, 42, 44, 46, 47
		6	not used	
		7	not used	
M-Bus Alarms	3	0	Communication error M-Bus channel 1	100
		1	Communication error M-Bus channel 2	110
		2	Communication error M-Bus channel 3	120
		3	Communication error M-Bus channel 4	130
		4	Fraud attempt M-Bus channel 1	103
		5	Fraud attempt M-Bus channel 2	113
		6	Fraud attempt M-Bus channel 3	123
		7	Fraud attempt M-Bus channel 4	133
Reserved	4 (MSB)	0	New M-Bus device discovered channel 1	105
		1	New M-Bus device discovered channel 2	115
		2	New M-Bus device discovered channel 3	125
		3	New M-Bus device discovered channel 4	135
		4	not used	
		5	not used	
		6	not used	
		7	not used	

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 CS 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 simplified status code is used for every entry. It can only be used for profiles containing cumulative values.

AMR Profile Status Code E meter (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.10.2.255**	R	R	
2	Value	Unsigned	AMR Profile Status Code E-meter	R	R	
	Specific methods	***m/o***				

AMR Profile Status Code M-Bus (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-x:96.10.3.255** **(x=channel number (1..4))**	R	R	
2	Value	Unsigned	AMR Profile Status Code M-Bus channel x	R	R	
	Specific methods	***m/o***				

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. At the same time the DNV flag is set because the capture period deviates from its nominal lengths and shall 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 can 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. At 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. At the same time, the DNV bit is set.

5 ABSTRACT OBJECTS

5.1 SAP assignment, Association LN, Security Setup, COSEM logical device name

SAP Assignment (Class ID: 17) Information about the logical devices in the physical device				P	M	Pr
1	Logical name	Octet-string	**0-0:41.0.0.255**	R	R	
2	Value	asslist_type	Only 1 logical device: The management logical device	R	R	
	Specific methods	***m/o***				
1	connect_logical_device (data)	o				

Association LN (Class ID: 15)				P	M	Pr
1	logical_name	octet-string	**0-0:40.0.0.255**	R	R	
2	object_list	objlist_type		R	R	
3	associated_partners_id	associated_partners_type		R	R	
4	application_context_ name	application_context_name		R	R	
5	xDLMS_context_info	xDLMS_context_type		R	R	
6	authentication_ mechanism_name	mechanism_name		R	R	
7	Secret	octet-string (minimum length = 16)			R	
8	association_status	enum		R	R	
9	security_setup_reference	octet-string	0-0:43.0.0.255	R	R	
	Specific methods	***m/o***				
1	reply_to_HLS_ authentication	m			X	
2	change_HLS_secret (data)[3]	m			X	
3	add_object (data)	o				
4	remove_object (data)	o				

[3] The new HLS secret is keywrapped. The key wrapping algorithm is as specified by the security suite. The KEK is the master key. The minimum length of the HLS secret is equal or bigger than the masterkey (i.e. 16 octets).

Security Set up (Class Id 64)				P	M	Pr
1	logical_name	octet-string	**0-0:43.0.0.255**		R	
2	security_policy	enum			R	
3	security_suite	enum	Value 0.		R	
4	client_system_title	octet-string			R	
5	server_system_title	octet-string			R	
	Specific methods	***m/o***				
1	security_activate	m			X	
2	global_key_transfer[4]	m			X	

COSEM Logical Device Name (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:42.0.0.255**	R	R	
2	Value	Octet-string	Unique identification of the logical device	R	R	
	Specific methods	***m/o***				

5.2 Identification numbers

Device ID 1 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.1.0.255**	R	R	
2	Value	Octet-string[16]	E-meter serial number (Serial number of the device, handled by the manufacturer)	R	R	
	Specific methods	***m/o***				

Device ID 2 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.1.1.255**	R	R	
2	Value	Octet-string[48]	E-meter equipment identifier (KEMA Code, serial number, last two digits of year of production (in total 17 characters))	R	R	
	Specific methods	***m/o***				

[4] The new key becomes effective with the next call from the CS within the same communication session. This means that the ACK will be encrypted with the old key. The next message shall use the new key.

Device ID 4 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.1.3.255**	R	R	
2	Value	Octet-string[48]	Location information (Owned and handled by the utility, has no meaning to the device.)	R	RW	
	Specific methods	***m/o***				

Device ID 5 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.1.4.255**	R	R	
2	Value	Octet-string[48]	Operational Hardware Identifier (The version identifier of the hardware in the meter, handled by the manufacturer).	R	R	
	Specific methods	***m/o***				

Device ID 8 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.1.7.255**	R	R	
2	Value	Octet-string[48]	Device initial hardware, software and configuration information	R	RW	
	Specific methods	***m/o***				

Device ID 9 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.1.8.255**	R	R	
2	Value	Octet-string[48]	Grid operators device ordering information	R	R	
	Specific methods	***m/o***				

5.3 Clock

Clock (Class ID: 8)				P	M	Pr
1	Logical name	Octet-string	**0-0:1.0.0.255**	R	R	
2	time	octet-string	current local date and time	R	RW	
3	time_zone	long	Value = -60		RW	
4	status	clock_status			R	
5	daylights_savings_begin	octet-string	last Sunday in March at 02:00		RW	
6	daylights_savings_end	octet-string	last Sunday in October at 03:00		RW	
7	daylights_savings_ deviation	integer	Value = 60		R	
8	daylights_savings_ enabled	boolean	Value = 1		RW	
9	clock_base	enum	Value = 1. Internal crystal		R	
	Specific methods	***m/o***				
1	adjust_to_quarter (data)	o				
2	adjust_to_measuring_ period (data)	o				
3	adjust_to_minute (data)	o				
4	adjust_to_preset_time (data)	o				
5	preset_adjusting_time (data)	o				
6	shift_time (data)	o				

Clock Time Shift Limit (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:0.9.11.255**		R	
2	value	unsigned	Value = 60. Maximum allowed time shift without registration of a time shift event		RW	
3	Scaler_unit	scal_unit_type	Value = {0,7}. Scaler=0, unit=seconds		R	
	Specific methods	***m/o***				
1	Reset (data)	o				

5.4 Activity Calendar and Special Days Table

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

NB: The order in which attributes 6 thru 9 are written by the CS is not relevant.

Special Days Table (Class ID: 11)				P	M	Pr
1	Logical name	Octet-string	**0-0:11.0.0.255**		R	
2	Entries	array[30]	Maximum of 30 special days		RW	
	Specific methods	***m/o***				
1	insert (data)	m			X	
2	delete (data)	m			X	

Currently active tariff (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**0-0:96.14.0.255**		R	
2	Value	Octet-string[2]	currently active tariff = active_mask of register activation object		R	
	Specific methods	***m/o***				

Tariffication Script Table (Class ID: 9)				P	M	Pr
1	Logical name	Octet-string	**0-0:10.0.100.255**		R	
2	Scripts	Array[2]	activation of certain tariff conditions service		RW	
	Specific methods	***m/o***				
1	Execute (data)	m	Long-unsigned		X	

Register Activation (Class ID: 6)				P	M	Pr
1	Logical name	Octet-string	**0-0:14.0.0.255**		R	
2	Register_assignment	Array[48]	Specifies an ordered list of COSEM objects assigned to the "Register activation" object		RW	
3	Mask_list	Array	Specifies a list of register activation masks - array register_act_mask (Writeable, size=8) - index_array := array unsigned (Writeable, size=64)		RW	
4	Active_mask	Octet-string	Defines the currently active mask		RW	
	Specific methods	***m/o***				
1	add_register (data)	m			X	
2	add_mask (data)	m			X	
3	delete_mask (data)	m			X	

5.5 Billing periods and profiles

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

Data of billing period 1 (Class ID: 7) **Monthly billing values**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:98.1.0.255**		R	
2	buffer	array			R	

3	capture_objects[5]	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};
			{4,0-1:24.2.1.255,2,0};
			{4,0-1:24.2.1.255,5,0};
			{4,0-2:24.2.1.255,2,0};
			{4,0-2:24.2.1.255,5,0};
			{4,0-3:24.2.1.255,2,0};
			{4,0-3:24.2.1.255,5,0};
			{4,0-4:24.2.1.255,2,0};
			{4,0-4:24.2.1.255,5,0};
			
			(= clock;+A rate 1;+A rate 2;−A rate 1;−A rate 2; 4 M-Bus register values & capture times of the M-Bus registers by the M-Bus devices)
			Can be extended with additional tariff registers
					R	

4	capture_period	double-long-unsigned	0, triggered from single action scheduler with billing period 1		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	13 months		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

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

Error Object (Class ID: 1) **Error register**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:97.97.0.255**		R	
2	Value	double-long-unsigned	Error code (See definition of error codes in paragraph 4.2.3)		RW	
	Specific methods	***m/o***				

[5] The value of a captured object may be replaced by “null-data” if it can be unambiguously recovered from the previous value (for example for time: if it can be calculated from the previous value and capture_period; or for a value: if it is equal to the previous value

Alarm Object (Class ID: 1) **Alarm register**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:97.98.0.255**		R	
2	Value	double-long-unsigned	Alarm code (See definition of alarm codes in paragraph 4.2.4)		RW	
	Specific methods	***m/o***				

Alarm filter (Class ID: 1)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:97.98.10.255**		R	
2	Value	double-long-unsigned	This filter defines the selection of events that are treated as alarms. Bit mask following the structure of the alarm register (See definition of alarm filters in paragraph 4.2.4) A “0” means that the event will not be treated as an alarm		RW	
	Specific methods	***m/o***				

5.7 Event Handling (events and logs)

Event Log (Class ID: 7) **Standard event log containing errors and alarms**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:99.98.0.255**		R	
2	buffer	array			R	

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0};
			{1,0-0:96.11.0.255,2,0}
			(= clock;event code)
			(See definition of event codes in paragraph 4.2.1)
					R	

4	capture_period	double-long-unsigned	0, asynchronously		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	100		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

5.8 Fraud detection (event log)

Fraud Detection Log (Class ID: 7) **Event log containing all fraud detection events**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:99.98.1.255**		R	
2	buffer	array			R	

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)
					R	

4	capture_period	double-long-unsigned	0, asynchronously		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	30		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

5.9 Power Failure (counter, thresholds and event log)

Number of power failures in any phases (Class ID: 1) **(single and polyphase meters)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:96.7.21.255**		R	
2	Value	long-unsigned			R	
	Specific methods	***m/o***				

Number of long power failures in any phases (Class ID: 1) **(single and polyphase meters)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:96.7.9.255**		R	
2	Value	long-unsigned			R	
	Specific methods	***m/o***				

Time threshold for long power failure (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:96.7.20.255**		R	
2	Value	long-unsigned	Value = 180		RW	
3	scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds		R	
	Specific methods	***m/o***				
1	reset (data)	o				

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

Power Failure Event Log (Class ID: 7)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:99.97.0.255**		R	
2	buffer	array			R	

3	capture_objects	Array [2]	{8,0-0:1.0.0.255,2,0};
			{3,0-0:96.7.19.255,2,0}
			(= timestamp; duration of long power failures in any phase)
			Timestamp = end of power failure
					R	

4	capture_period	double-long-unsigned	0, asynchronously		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	10		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

5.10 P1 objects (messages, readout list)

Consumer Message Text (Class ID: 1) **Consumer message text sent to port P1**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:96.13.0.255**		R	
2	Value	Octet-string[1024]	Message text sent to port P1 without any further interpretation		RW	
	Specific methods	***m/o***				

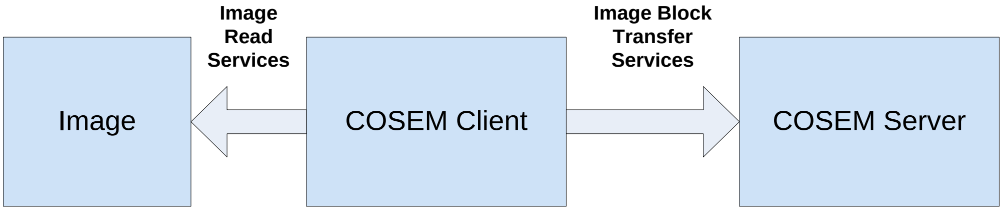
Consumer Message Code (Class ID: 1) **Consumer message code shown on display and P1**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:96.13.1.255**		R	
2	Value	Octet-string[8]	Message code must be shown on numeric display (including scroll feature).		RW	
	Specific methods	***m/o***				

The Standard Readout Object List is shown in P3, Annex B.

General local port readout (Class ID: 7) P1 port readout list				P	M	Pr
1	Logical name	Octet-string	**0-0:21.0.0.255**		R	
2	buffer	array	last readout		R	
3	capture_objects	Array [64]	readout objects, a maximum of 64 entries is possible		RW	
4	capture_period	double-long-unsigned	10, update period of P1 [s]		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	1		R	
	Specific methods	***m/o***				
1	reset ()	m				
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

5.11 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.



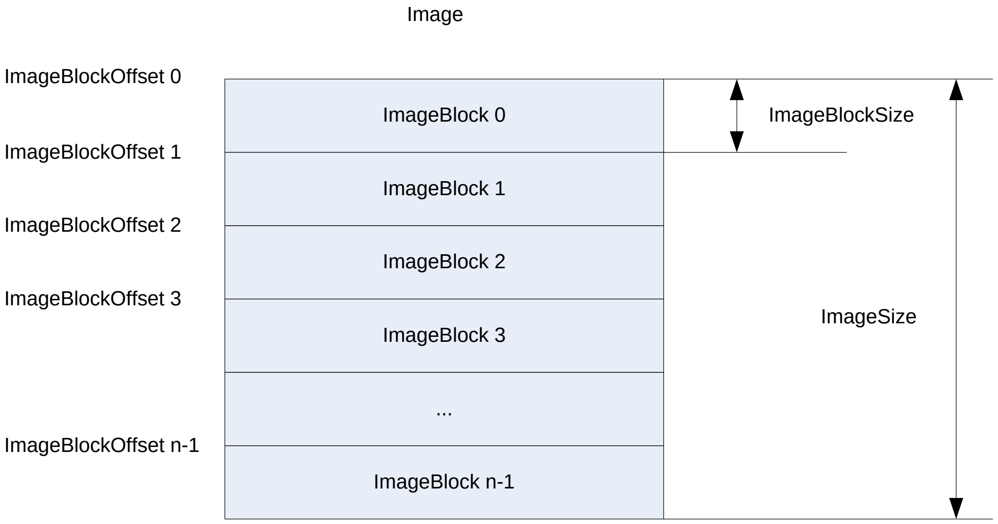


Image is data of specified size.

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**				**P**	**M**	**Pr**
1	Logical_name	octet-string	**0-0:44.0.0.255**		R	
2	Image_block_size	double-long-unsigned			R	
3	Image_transferred_blocks_status	bitstring			R	
4	Image_first_ transferred_number	double-long-unsigned			R	
5	Transfer_enabled	boolean			RW	
6	Images_transfer_status	enum			R	
7	Image_to_activate_info	array			R	
	Specific methods	***m/o***				
1	Image_transfer_initiate	m			X	
2	image_block_transfer	m			X	
3	Image_verify	m			X	
4	Image_activate	m			X	

The firmware upgrade process is described in detail in the Bluebook version 10.

As part of step 2, the Image_to Activate_info array shall be reset.

In step 5, image verification shall be started by the server immediately after the last block has been received and be based on calculation of a CRC32 hash value of the firmware image data.

The first block of data that is transferred, shall contain a hash value that is calculated over the firmware CRC32 image data only (excluding the hash value itself).

After the firmware has successfully been activated, the E meter performs a self check and logs the results. The results of the activation can be retrieved from the image_transfer_status attribute.

The image shall be activated by a specific command from the client, immediately after the current application association has ended

In step 7 the image activation shall only be possible if the image transfer status is 3 (Image verification successful.

Additionally to the process described in the DLMS Blue Book version 10, as a final step the E meter performs a self-check and logs whether the firmware upgrade was activated successfully.

5.12 **P0 Local port Set-up**

IEC local port setup objects (class_id: 19) OBIS code for IEC optical port setup: 0-b:20.0.0.255

COSEM Local port setup object (Class ID: 19)			**P**	**M**	**Pr**
Logical name	octet-string	**0-0:20.0.0.255**		R	
Default_mode	enum	Value = 0 (protocol according to IEC 62056-21)		R	
Default_baud	enum	Value = 5 (9600 baud)		R	
Prop_baud	enum	Value = 5 (9600 baud)		R	
Response_time	enum	Value = 200		R	
Device_addr	octet-string	Value = 17		R	
Pass_p1	Octet-string	Not used. DLMS security is used in IEC 62056-21 mode E		R	
Pass_p2	Octet-string	Not used. DLMS security is used in IEC 62056-21 mode E		R	
Pass_p3	Octet-string	Not used. DLMS security is used in IEC 62056-21 mode E		R	
Specific methods	***m/o***				

5.15 **IEC HDLC Set up**

IEC HDLC setup object (Class ID: 23)				**P**	**M**	**Pr**
1	Logical name	octet-string	**0-0:22.0.0.255**		R	
2	Comm_speed	enum	Value = 5 (9600 baud)		R	
3	Window_size_transmit	unsigned	Value = 1		R	
4	Window_size_receive	unsigned	Value = 1		R	
5	max_info_field_length_ transmit	long-unsigned	Value = 128		R	
6	max_info_field_length_ receive	long-unsigned	Value = 128		R	
7	inter_octet_time_out	long-unsigned	Value = 25		R	
8	inactivity_time_out	long-unsigned	Value = 120		R	
9	device_address	long-unsigned	Value = 17		R	
	Specific methods	***m/o***				

5.16 Display readout modes

The display read-out list objects in this section define which OBIS-codes must be displayed in which display mode. The actual list of OBIS-codes that must be displayed is defined in annex B. The values are not stored in the Display Readout Objects.

General display readout (Class ID: 7) **Auto scroll readout list**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:21.0.1.255**		R	
2	buffer	array	last readout		R	
3	capture_objects	Array [64]	readout objects, a maximum of 64 entries. See Annex B for the items in the list		RW	
4	capture_period	double-long-unsigned	10, update period [s]		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	1		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

Alternate display readout (Class ID: 7) **Manual scroll readout list**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:21.0.2.255**		R	
2	buffer	array	last readout		R	
3	capture_objects	Array [64]	readout objects, a maximum of 64 entries. See Annex B for the items in the list		RW	
4	capture_period	double-long-unsigned	10, update period [s]		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	1		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

Service display readout (Class ID: 7) **Service mode readout list**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-0:21.0.3.255**		R	
2	buffer	array	last readout		R	
3	capture_objects	Array [64]	readout objects, a maximum of 64 entries. See Annex B for the items in the list		RW	
4	capture_period	double-long-unsigned	10, update period [s]		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	1		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

6 ELECTRICITY RELATED OBJECTS

This section holds a complete overview of all electricity related objects and their attributes. In Annex A the minimal numbers of digits used throughout the whole metering chain are shown.

6.1 Identification numbers (i.e. firmware version)

MID requires a certain format for the identification of the metrological part of the Firmware. For this the following is applicable:

OBIS-object 1-0:0.2.0.255

Active Firmware Identifier; should be a more or less "readable" identification of the FW version, for example: "1.0" or “15”

OBIS-object 1-0:0.2.8.255

Active Firmware Signature; should be the Hash code of this version, for example: “123AF46558283F32…”

Active firmware version (Class ID: 1)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:0.2.0.255**	R	R	
2	Value	Octet-string	Active Firmware Identifier	R	R	
	Specific methods	***m/o***				

Active firmware version (Class ID: 1)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:0.2.8.255**	R	R	
2	Value	Octet-string	Active Firmware Signature	R	R	
	Specific methods	***m/o***				

Active firmware version (Class ID: 1)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-1:0.2.0.255**	R	R	
2	Value	Octet-string	Module Active Firmware Identifier	R	R	
	Specific methods	***m/o***				

Active firmware version (Class ID: 1)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-1:0.2.8.255**	R	R	
2	Value	Octet-string	Module Firmware Signature	R	R	
	Specific methods	***m/o***				

Active firmware version (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**1-2:0.2.0.255**	R	R	
2	Value	Octet-string	Communication Module Active Firmware Identifier	R	R	
	Specific methods	***m/o***				

Active firmware version (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**1-2:0.2.8.255**	R	R	
2	Value	Octet-string	Communication Module Firmware Signature	R	R	
	Specific methods	***m/o***				

P1 port DSMR version (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	1-3:0.2.8.255	R	R	
2	Value	Octet-string	P1 port DSMR version	R	R	
	Specific methods	***m/o***				

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

Active energy import (+A) (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:1.8.0.255**		R	
2	Value	double-long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh		R	
	Specific methods	***m/o***				
	reset (data)	o				

Active energy export –A (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:2.8.0.255**		R	
2	Value	double-long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh		R	
	Specific methods	***m/o***				
	reset (data)	o				

Active energy import (+A) rate 1 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:1.8.1.255**		R	
2	Value	double-long-unsigned	Rate 1 = Low tariff		R	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh		R	
	Specific methods	***m/o***				
	reset (data)	o				

Active energy import (+A) rate 2 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:1.8.2.255**		R	
2	Value	double-long-unsigned	Rate 2 = normal/high tariff		R	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh		R	
	Specific methods	***m/o***				
	reset (data)	o				

Active energy export (-A) rate 1 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:2.8.1.255**		R	
2	Value	double-long-unsigned	Rate 1 = Low Tariff		R	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh		R	
	Specific methods	***m/o***				
	reset (data)	o				

Active energy export (-A) rate 2 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:2.8.2.255**		R	
2	Value	double-long-unsigned	Rate 2 = Normal/high tariff		R	
3	Scaler_unit	scal_unit_type	Value = {0,30}, scaler=0, unit=Wh		R	
	Specific methods	***m/o***				
	reset (data)	o				

6.3 E profiles (15', daily)

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

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0};
			{1,0-0:96.10.2.255,2,0};
			{3,1-0:1.8.0.255,2,0};
			{3,1-0:2.8.0.255,2,0}
			(= clock; AMR profile status; +A;−A)
			
			Profile status ➔ see paragraph 4.2.5
					R	

4	capture_period	double-long-unsigned	900 (15 minutes)		R	
5	sort_method	enum	1 or 3 (unsorted (FIFO) or sorted (largest))		R	
6	sort_object	object definition	none or {8,0-0:1.0.0.255,2,0} (unsorted or sorted by clock)		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	(10 days)		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

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

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; clock {1,0-0:96.10.2.255,2,0} AMR profile status {3,1-0:1.8.1.255,2,0}; +A rate1 {3,1-0:1.8.2.255,2,0} +A rate 2 {3,1-0:2.8.1.255,2,0}; -A rate1 {3,1-0:2.8.2.255,2,0} -A rate 2 {4,0-0.1.24.2.1.255,2,0}, M-Bus Master Value 1 Channel 1 {4,0-0.1.24.2.1.255,5,0}, M-Bus Master Value 1 Channel 1 Capture time of the M-Bus registers by the M-Bus devices) {4,0-0.2.24.2.1.255,2,0}, M-Bus Master Value 1 Channel 2 {4,0-0.2.24.2.1.255,5,0}, M-Bus Master Value 1 Channel 2 Capture time of the M-Bus registers by the M-Bus devices) {4,0-0.3.24.2.1.255,2,0}, M-Bus Master Value 1 Channel 3 {4,0-0.3.24.2.1.255,5,0}, M-Bus Master Value 1 Channel 3 Capture time of the M-Bus registers by the M-Bus devices) {4,0-0.4.24.2.1.255,2,0) M-Bus Master Value 1 Channel 4 {4,0-0.4.24.2.1.255,5,0}, M-Bus Master Value 1 Channel 4 Capture time of the M-Bus registers by the M-Bus devices) AMR Profile status see paragraph 4.2.5[6]		R	
4	capture_period	double-long-unsigned	86400 (daily)		R	
5	sort_method	enum	1 or 3 (unsorted (FIFO) or sorted (largest))		R	
6	sort_object	object definition	none or {8,0-0:1.0.0.255,2,0} (unsorted or sorted by clock)		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	40 (40 days)		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

[6] This represents the combined statuses of all the devices. To identify the faulty device it is necessary to read the interval data of the different meters. This can only be done if the customer allows the reading of interval data.)

6.4 Instantaneous and Average Values

Instantaneous voltage L1 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:32.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V		RW	
	Specific methods	***m/o***				
	reset (data)	o				

Average voltage L1 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:32.24.0.255**		R	
2	Value	long-unsigned	10 minutes average voltage. Averaging scheme 3 is used.		R	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous current L1 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:31.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A		RW	
	Specific methods	***m/o***				
	reset (data)	o				

Average active power (+P) L1 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:21.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average active power (-P) L1 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:22.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power L1 (+P) (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:21.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power L1 (-P) (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:22.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average reactive power (+Q) L1 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:23.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,29}, scaler=0, unit=var		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average reactive power (-Q) L1(Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:24.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,29}, scaler=0, unit=var		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average current L1(Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:31.24.0.255**		R	
2	Value	long-unsigned	10 minutes average current. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous voltage L2 (Class ID: 3) (polyphase meters only)				P	M	Pr
1	Logical name	Octet-string	**1-0:52.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V		RW	
	Specific methods	***m/o***				
	reset (data)	o				

Average voltage L2 (Class ID: 3) (polyphase meters only)				P	M	Pr
1	Logical name	Octet-string	**1-0:52.24.0.255**		R	
2	Value	long-unsigned	10 minutes average voltage. Averaging scheme 3 is used.		R	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous current L2 (Class ID: 3) (polyphase meters only)				P	M	Pr
1	Logical name	Octet-string	**1-0:51.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A		RW	
	Specific methods	***m/o***				
	reset (data)	o				

Average active power (+P) L2 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:41.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average active power (-P) L2 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:42.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power L2 (+P) (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:41.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power L2 (-P) (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:42.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average reactive power (+Q) L2 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:43.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,29}, scaler=0, unit=var		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average reactive power (-Q) L2 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:44.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,29}, scaler=0, unit=var		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average current L2(Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:51.24.0.255**		R	
2	Value	long-unsigned	10 minutes average current. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous voltage L3 (Class ID: 3) **(polyphase meters only)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:72.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V		RW	
	Specific methods	***m/o***				
	reset (data)	o				

Average voltage L3 (Class ID: 3) **(polyphase meters only)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:72.24.0.255**		R	
2	Value	long-unsigned	10 minutes average voltage. Averaging scheme 3 is used.		R	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V, resolution: 0 V		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous current L3 (Class ID: 3) **(polyphase meters only)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:71.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A		RW	
	Specific methods	***m/o***				
	reset (data)	o				

Average active power (+P) L3 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:61.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average active power (-P) L3 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:62.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power L3 (+P) (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:61.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power L3 (-P) (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:62.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average reactive power (+Q) L3 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:63.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,29}, scaler=0, unit=var		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average reactive power (-Q) L3 (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:64.24.0.255**		R	
2	Value	long-unsigned	10 minutes average power. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,29}, scaler=0, unit=var		R	
	Specific methods	***m/o***				
	reset (data)	o				

Average current L3 (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:71.24.0.255**		R	
2	Value	long-unsigned	10 minutes average current. Averaging scheme 3 is used		R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power (+P) (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:1.7.0.255**		R	
2	Value	Double long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous active power (-P) (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:2.7.0.255**		R	
2	Value	long-unsigned			R	
3	Scaler_unit	scal_unit_type	Value = {0,27}, scaler=0, unit=W		R	
	Specific methods	***m/o***				
	reset (data)	o				

Instantaneous current (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:90.7.0.255**		R	
2	Value	long-unsigned	Total current (sum of all phases)		R	
3	Scaler_unit	scal_unit_type	Value = {0,33}, scaler=0, unit=A, resolution: 0 A		R	
	Specific methods	***m/o***				
	reset (data)	o				

6.5 Power Quality (Voltage sags and swells)

Threshold for voltage sag (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:12.31.0.255**		R	
2	Value	long-unsigned	Value = 207, threshold for the detection of power sags, programmable according to requirements of the GO		RW	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V		R	
	Specific methods	***m/o***				
	reset (data)	o				

Time threshold for voltage sag (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:12.43.0.255**		R	
2	Value	long-unsigned	Value = 30, duration of the voltage sag until it is detected, programmable according to requirements of the GO		RW	
3	Scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds		R	
	Specific methods	***m/o***				
	reset (data)	o				

Number of voltage sags in phase L1 (Class ID: 1)				P	M	Pr
1	Logical name	Octet-string	**1-0:32.32.0.255**		R	
2	Value	long-unsigned	Number of voltage sags		R	
	Specific methods	***m/o***				

Number of voltage sags in phase L2 (Class ID: 1) (polyphase meters only)				P	M	Pr
1	Logical name	Octet-string	**1-0:52.32.0.255**		R	
2	Value	long-unsigned	Number of voltage sags		R	
	Specific methods	***m/o***				

Number of voltage sags in phase L3 (Class ID: 1) (polyphase meters only)				P	M	Pr
1	Logical name	Octet-string	**1-0:72.32.0.255**		R	
2	Value	long-unsigned	Number of voltage sags		R	
	Specific methods	***m/o***				

Threshold for voltage swell (Class ID: 3)				P	M	Pr
1	Logical name	Octet-string	**1-0:12.35.0.255**		R	
2	Value	long-unsigned	Value = 253, threshold for the detection of power swells, programmable according to requirements of the GO		RW	
3	Scaler_unit	scal_unit_type	Value = {0,35}, scaler=0, unit=V		R	
	Specific methods	***m/o***				
	reset (data)	o				

Time threshold for voltage swell (Class ID: 3)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:12.44.0.255**		R	
2	Value	long-unsigned	Value = 30, duration of the voltage swell until it is detected, programmable according to requirements of the GO		RW	
3	Scaler_unit	scal_unit_type	Value = {0,7}, scaler=0, unit=seconds		R	
	Specific methods	***m/o***				
	reset (data)	o				

Number of voltage swells in phase L1 (Class ID: 1)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:32.36.0.255**		R	
2	Value	long-unsigned	Number of voltage swells		R	
	Specific methods	***m/o***				

Number of voltage swells in phase L2 (Class ID: 1) **(polyphase meters only)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:52.36.0.255**		R	
2	Value	long-unsigned	Number of voltage swells		R	
	Specific methods	***m/o***				

Number of voltage swells in phase L3 (Class ID: 1) **(polyphase meters only)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**1-0:72.36.0.255**		R	
2	Value	long-unsigned	Number of voltage swells		R	
	Specific methods	***m/o***				

7 M-BUS RELATED OBJECTS

7.1 M-Bus Setup

M-Bus Client Setup (Class ID: 72) **Setup in M-Bus master for every M-Bus client (4 instances, one per channel, see additional info)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-x:24.1.0.255**[7]		R	
2	mbus_port_reference	Octet-string			R	
3	capture_definition	array			R[8]	
4	capture_period	double-long-unsigned			RW	
5	primary_address	unsigned			RW	
6	identification_number	double-long unsigned interpreted as BCD[9]	(Only last 8 digits of identiification number) Part of short ID		RW	
7	manufacturer_Id	long_unsigned	**Part of short ID**		RW	
8	version	unsigned	**Part of short ID**		RW	
9	device_type	unsigned	**Part of short ID**		RW	
10	access_number	unsigned			R	
11	status	unsigned			R	
12	alarm	unsigned			R	
13	Configuration	long-unsigned	Default value is 99 (unknown), 0 or 15		R	
14	encryption_key_status	enum	Carries the status of the encryption key [10] Enum: (0) no encryption_key (1) encryption_key set (2) encryption_key transferred (3) encryption_key set and transferred (4) encryption_key in use		R	
	Specific methods	***m/o***				
1	slave_install	m			X	
2	slave_deinstall	m			X	
3	capture	m			X	
4	reset_alarm	m			X	
5	synchronize_clock	m			X	
6	data_send	m			X	
7	set_encryption_key	m			X	
8	transfer_key	m			X	

[7] A new DLMS channel will be assigned to each new device, x=channel number 1..4

[8] Capture definition is determined by E-Meter dynamically from received M-Bus telegrams. Central System can not overrule this definition.

[9] See TC294_N0280_Updated_draft_prEN_13757-3_for_launch_of_pub

[10] The states are based on the methods 7 and 8 that are part of the object 'M-bus client object' as defined in the bluebook. For example the status 'encryption_key_set and transferred' means that both methods 7 and 8 are invoked.

The encryption status can be read out from the configuration word of any message from the M-Bus device and will be stored in attribute 13. The encryption value can either be 0 (not encrypted) or 15 (encrypted using AES).

The version byte indicates the DSMR compliancy level and is stored in attribute 8. This is relevant to deduct the method used for frame counters. To be able to fully deduct the status of the encryption on P2, the receptions of new keys for P2 as well as the keys forwarded to the M-Bus device are relevant.

Slave_deinstall:

De-installs the slave device. The main purpose of this service is to uninstall the M-Bus slave device and to prepare the master for the installation of a new device.
The following actions are performed:

- the M-Bus address is set to 0 in the M-Bus slave device;
- the encryption key transferred previously to the M-Bus slave device is destroyed; the master key is not affected.
- the attribute primary_address is also set to 0.
- the attribute configuration should be set to 99
- the attribute encryption_key_status should be set to 0

M-Bus master port setup (Class ID 74)				**P**	**M**	**Pr**
1	logical_name	octet-string	0-x:24.6.0.255		R	
2	comm_speed	enum	Value = 3; 2400 baud		R	
	Specific methods	***m/o***				

7.2 Identification numbers

Device ID 1 (Class ID: 1) **M-Bus Equipment identifier (4 instances, one per channel)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-x:96.1.0.255** (x=channel number (1..4))		R	
2	Value	Octet-string[48]	M-Bus long header		R	
	Specific methods	***m/o***				

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

1. 0.1.96.1.0.255 - CH1
2. 0.2.96.1.0.255 - CH2
3. 0.3.96.1.0.255 - CH3
3. 0.4.96.1.0.255 - CH4

7.3 Registers

7.3.1 *Measurement value*

This Register is used to make the M-Bus device measured value available.

M-Bus Master Value (Class ID: 4) **Instance specific (4 instances, one per channel)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-x:24.2.1.255** (x=channel number (1..4), One channel per M-Bus device		R	
2	Value	double-long-unsigned	Measurement value		R	
3	scaler_unit	scal_unit_type	set by the E-meter by reading the M-Bus device VIF/DIF combination from the M-Bus device		R	
4	status	octet-string	status of M-Bus device		R	
5	capture_time	octet-string	time of last successful readout		R	
	Specific methods	***m/o***				
1	reset (data)	o				

7.3.2 *M-Bus-Device configuration*

This register is used to make the M-Bus device configuration available. The M-Bus-Device configuration consists of:

- Model / Version (VIF/VIFE = FDh 0Ch)
- Hardware version number, (VIF/VIFE = FDh 0Dh)
- Metrology (firmware) version number, (VIF/VIFE = FDh 0Eh)
- Other software version number, (VIF/VIFE = FDh 0Fh)
- Meter Configuration (including “valve device present” (VIF/VIFE = FDh 67h)

M-Bus Device configuration (Class ID: 4) Instance specific (4 instances, one per channel)				P	M	Pr
1	Logical name	Octet-string	**0-x:24.2.2.255** (x=channel number (1..4), One channel per M-Bus device		R	
2	Value	octet-string[255]	String with concatenation of 5 (variable length) information fields: [Model/version] [Hardware version number] [Metrology (firmware) version number] [Other software version number] [Meter Configuration] Each fields is to be terminated with CR/LF (ASCII characters <CR><LF>) The first 4 information fields have a maximum length of 61 characters each and contain the info as received from the M-Bus. If the information from the M-Bus is too long then the leftmost octets are skipped. The last information field is 1 character (8 bits as received from M-Bus device).		R	
3	scaler_unit	scal_unit_type	Value = {0,255}, scaler = 0, unit = count		R	
4	status	octet-string	status of M-Bus device		R	
5	capture_time	octet-string	Time of last successful readout. The M-Bus device configuration is (at least) read by the E-Meter from the M-Bus device at the end of the M-Bus device installation procedure.		R	
	Specific methods	***m/o***				
1	reset (data)	o				

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)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-x:24.3.0.255** (x=channel number (1..4))		R	
2	buffer	array	The buffer must be filled monotonously, i.e. no irregular entries are allowed		R	

3	capture_objects	Array	{8,0-0:1.0.0.255,2,0};
			{1,0-x: 96.10.3.255,2,0}
			{4,0-x:24.2.1.255,2,0}
			{4,0-x:24.2.1.255,5,0}
			(x=channel number (1..4))
			(=clock; AMR profile status; M-Bus master value object & capture times of the M-Bus registers by the M-Bus devices)
			
			AMR profile status see paragraph 4.2.5
					R	

4	capture_period	double-long-unsigned	3600, every hour		R	
5	sort_method	enum	1 or 3 (unsorted (FIFO) or sorted (largest))		R	
6	sort_object	object definition	none or {8,0-0:1.0.0.255,2,0}(unsorted or sorted by clock)		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	240 (10 days)		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

7.5 Event Log (Class id = 7)

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

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
					R	

4	capture_period	double-long-unsigned	0, asynchronously		R	
5	sort_method	enum	1, unsorted (FIFO)		R	
6	sort_object	object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	≥30		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

7.6 M-Bus Master Control log

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)**				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-x:24.5.0.255** (x=channel number (1..4))		R	
2	buffer	Array			R	
3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; {1,0-x: 96.11.4.255,2,0} (x=channel number (1..4)) (=clock; control event code) Event codes must be defined (paragraph 4.2.1)		R	
4	capture_period	double-long-unsigned	0, asynchronously		R	
5	sort_method	Enum	1, unsorted (FIFO)		R	
6	sort_object	Object definition	None, unsorted		R	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	10		R	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m				
3	Reserved from previous versions					
4	Reserved from previous versions					

8 COUNTRY SPECIFIC OBJECTS

8.1 Administrative in/out

Administrative in/out P3 (Class ID: 1)				P	M	Pr
1	logical_name	Octet string	0-1:94.31.0.255		R	
2	value	enum	Value = 0: Undefined Value = 1: Meter is administrative off (Opt Out) Value = 2: Meter is administrative on (Default)		RW	
	Specific methods	***m/o***				

The administrative in/out object in the electricity meter (obis code 0-1:94.31.0.255) holds the attribute that determines the state the meter is in.

8.2 Connection watchdog timer

Connection watchdog timer P3 (Class ID: 1)				P	M	Pr
1	logical_name	Octet string	0-1:94.31.2.255		R	
2	value	Long-unsigned	Unit=Hours		RW	
	Specific methods	***m/o***				

The connection watchdog timer object in the electricity meter (obis code 0-1:94.31.2.255) holds an attribute with the value of the watchdog timer in hours. A watchdog timer makes sure that the modem is reset after a defined period of no contact with the CS.

8.3 Configuration object

The configuration object in the electricity meter (obis code 0-1:94.31.3.255) holds a number of attributes that determine the behaviour of the meter. The actual values of these attributes are set before operation starts (possibly at production time) and are grid operator dependent. The purpose of this configuration object is to adjust the meter to the differences in processes of the various grid operators.

When configuration settings are changed they can be applied during the current association or directly there after. But an application association must always be closed in a proper way.

Configuration object (Class ID: 1)				P	M	Pr
1	logical_name	Octet string	0-1:94.31.3.255		R	
2	value	Structure	See below		RW	
	Specific methods	***m/o***				

Attribute description
Value ::= structure
{

GPRS_operation_mode enum

Flags bitstring (16)

}

GPRS_operation_mode	Enum ::= {always_on == 1, triggered == 2)

Flags

discover_on_open_cover (bit 0 - MSB)	Indicates whether the M-Bus discovery process (see P2 companion standard) is started when the cover of the M-Bus connections on the electricity meter is removed.
discover_on_power_on (bit 1)	Indicates whether the M-Bus discovery process (see P2 companion standard) is started when the power to the electricity meter is switched on.
dynamic_mbus_address (bit 2)	Indicates whether the M-Bus device should use dynamic or static addressing. Dynamic addressing entails that the primary address of the M-Bus device is reset to 0 if the device is decommissioned.
P0_enable (bit 3)	Indicates whether communication via P0 is enabled or not. (disabled == 0, enabled ==1)
HLS_3_on_P3_enable (bit 4)	Indicates whether authentication via HLS method 3 is enabled on P3 (disabled == 0, enabled ==1)
HLS_4_on_P3_enable (bit 5)	Indicates whether authentication via HLS method 4 is enabled on P3 (disabled == 0, enabled ==1)
HLS_5_on_P3_enable (bit 6)	Indicates whether authentication via HLS method 5 is enabled on P3 (disabled == 0, enabled ==1)
HLS_3_on_P0_enable (bit 7)	Indicates whether authentication via HLS method 3 is enabled on P0 (disabled == 0, enabled ==1)
HLS_4_on_P0_enable (bit 8)	Indicates whether authentication via HLS method 4 is enabled on P0 (disabled == 0, enabled ==1)
HLS_5_on_P0_enable (bit 9 - LSB)	Indicates whether authentication via HLS method 5 is enabled on P0 (disabled == 0, enabled ==1)

8.4 GPRS Network Information

GPRS Network Information P3 (Class ID: 1)				P	M	Pr
1	logical_name	Octet string	0-1:94.31.4.255		R	
2	value	Structure	See below		R	
	Specific methods	***m/o***				

Attribute description
 Value ::= structure

```
{
        signal_strength              integer,
        number_of_base_stations      integer,
        network_id                   octetstring
}
```

signal_strength	Current signal strength in dBm for the currently connected base station
number_of_base_stations	The currently reachable number of base stations with medium and high signal strength levels.
network_id	

8.5 Meter Reset

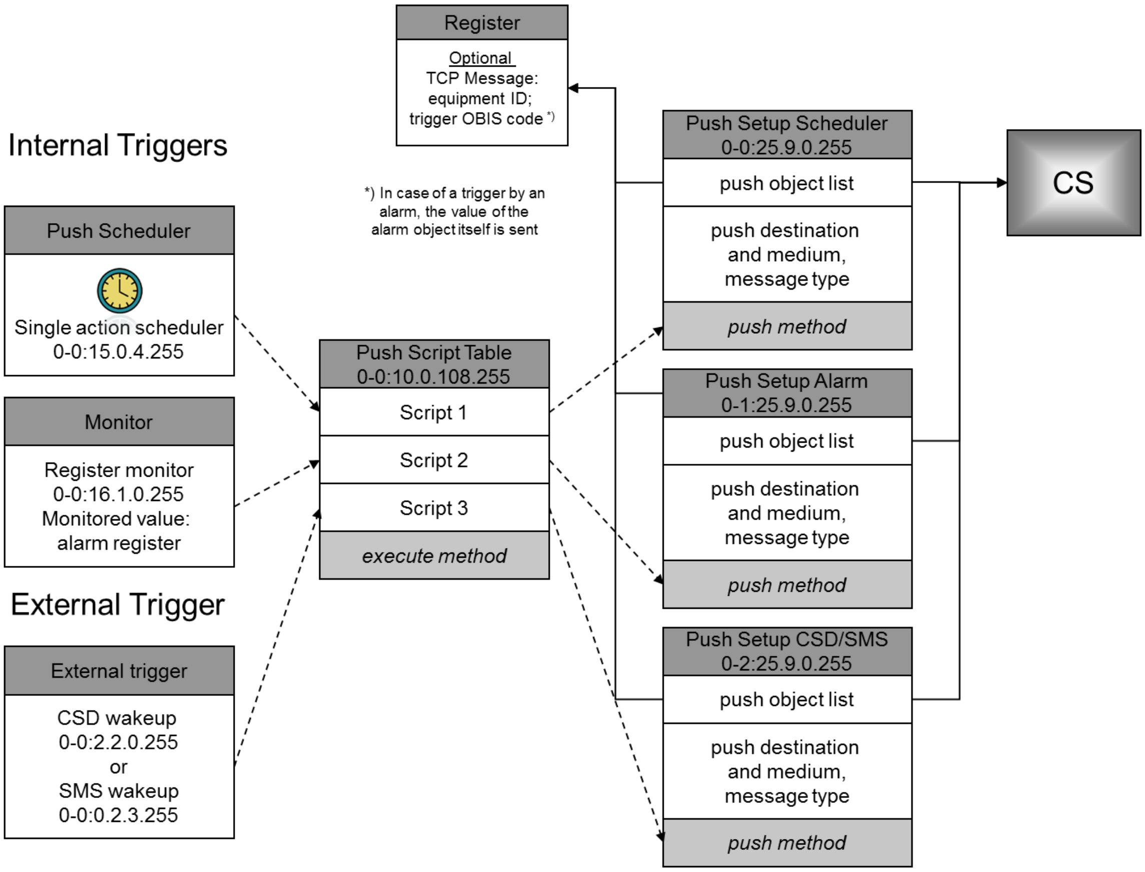
Meter Reset to Factory Setting (Class ID: 9)				P	M	Pr
1	Logical name	Octet-string	**0-1:94.31.5.255**		R	
2	Scripts	Array[1]	Script for Meter Reset to Factory Setting. Metrological Registers (1-0:1.8.1.255, 1-0:1.8.2.255, 1-0:2.8.1.255, 1-0:2.8.2.255) are not changed; other objects will be reset to factory setting. Also the security-policy is reset to factory setting. The changes will be done after the release of the active application association.		R	
	Specific methods	***m/o***				
1	Execute (data)	m	Long-unsigned		X	

8.6 Definable Load Profile

In this object a number of objects/attributes can be defined which values can be monitored, e.g. active/reactive power. The client will configure the object at runtime dependant on the measurement. The default configuration (attribute values listed in the table below) is not done for any specific measurement. The maximum buffer size is 960 entries with each a length of Octet-String(12) for the timestamp and in addition 20 double-long-unsigned entries for the measurement values. The capture period is independent of the measuring period of the capture objects e.g. the average voltage is measured with an interval of 10 minutes while the capture period can be another value

Definable load profile (Class ID: 7)				**P**	**M**	**Pr**
1	Logical name	Octet-string	**0-1:94.31.6.255**		R	
2	buffer	array			R	
3	capture_objects	Array	{8,0-0:1.0.0.255,2,0}; clock and other OBIS codes depending on configuration of grid operator		RW	
4	capture_period	double-long-unsigned	86400 (daily)		RW	
5	sort_method	enum	3 (1 = unsorted (FIFO) or 3 = sorted (largest))		RW	
6	sort_object	object definition	{8,0-0:1.0.0.255,2,0} (sorted by clock)		RW	
7	entries_in_use	double-long-unsigned			R	
8	profile_entries	double-long-unsigned	960 (max # of entries)		RW	
	Specific methods	***m/o***				
1	reset ()	m			X	
2	capture ()	m			X	
3	Reserved from previous versions					
4	Reserved from previous versions					

9 GPRS RELATED OBJECTS



The “Push setup” interface mainly contains a list of references of the object (attributes) to be pushed to the CS as well as the push destination and the communication medium to be used. A trigger (scheduler, monitor, wake-up call, etc.) calls a script entry in the push script table object which then invokes the push method of the related “Push setup” object. This method at the end handles the sending of the push data to the CS using the communication channel defined. The “Push setup” interface class also defines the communication time windows and the handling of retries for a push operation.

In the DSMR specification the “Push setup” interface is not used to send data to the CS but to initiate a connection. The Central System will retrieve the relevant data. However, as part of the communication initiation a message can be send to the CS which is defined in the push object list.

The following triggers can call a script entry in a push script table object are possible:

- Push scheduler.
  This is used for regular communication (see figure 1 in the DSMR GPRS Companion Standard).
- Monitor.
  This is used for alarming

- External trigger.
  This is used for CSD or SMS wake up mechanisms initiated from the CS

These triggers call a script entry in the Push Script Table object which then invokes the push method of the related “Push setup” object.

The push setup object takes care for connection management:

- starting the GPRS session
- setting up the PDP context
- optional sending the TCP message configured in the push object list containing the equipment identifier and the OBIS code of the trigger (see figure 2 in the DSMR GPRS Companion Standard)
- and preparing the TCP connection (open a server socket in listener mode) for the DLMS data transfer (initiated by the CS)

Note that the TCP connection, PDP context and GPRS attach are not destroyed after sending the objects defined in the push object list but only after all connections on the server socket are closed or timed out.

Some objects in this section use time windows (listening-window, calling-window) . The following settings shall be taken into account:

- A never active window is configured by setting the start_time to “unspecified” and a dummy end_time of 23:59:59:99,
- An always active window is configured by setting the start_time as empty (array[0]).

9.1 Triggers

9.1.1 *Internal triggering by Push Scheduler*

The meter can be triggered on a regular basis to setup a GPRS connection.. On the regular time, indicated by the Execution time (attribute 4) the associated script in attribute 2 is executed. This script is executed and the Push Setup object will trigger the connection management. In case this functionality is not used, the Execution time will be empty.

Single action scheduler (Class ID: 22, version=0)				**P**	**M**	**Pr**
1	logical_name	Octet string	0-0:15.0.4.255		R	
2	Executed_script	script	Contains the script which contains the push method invocation of the Push Setup Scheduler object.		RW	
3	Type	enum	Default value 1 is used.		RW	
4	Execution time	array	Contains the trigger moment.		RW	
	Specific methods	***m/o***				

9.1.2 *Internal triggering by Alarms*

The meter can be triggered by an alarm to setup a GPRS connection. An event can cause an alarm (depending on the alarm filter). If an alarm (monitored value) is raised the associated script in attribute 4 is executed. This script is executed and the Push Setup object will trigger the connection management. In case this functionality is not used, the Actions array will be empty.

Register monitor (Class ID: 21, version=0)				**P**	**M**	**Pr**
1	logical_name	Octet string	0-0:16.1.0.255		R	
2	Thresholds	array	Value of the alarm register when there are no alarms (this is 0). Note that the alarm register contains only filtered events.		R	
3	Value_definition	array	Obis code of the alarm register {1,0-0:97.98.0.255,2}		R	
4	Actions	array	The action_up contains the script which contains the push method invocation of the Push Setup Alarm object		RW	
	Specific methods	***m/o***				

The Register Monitor attribute “Actions” defines the action when the content of “Monitored-Value” crosses the “Thresholds” in the upwards direction.
In order to prevent that executing of the script is aborted because of a power-down, the comparing of the monitored value is done everytime the meter is powered up. This will results in the triggering the “Actions” again when one or more alarms in the alarm register are still active after a power down.

9.1.3 *External trigger: wakeup using CSD call*

The meter can be triggered by a Circuit Switched Data (CSD) call to setup a GPRS connection. The CSD call will be answered with a call reject. The listening window is active in case CSD wakeup is used. If the calling number is in the list_of_callers_and_actions, the associated script is executed. This script is executed and the Push Setup object will trigger the connection management.

See annex C for the Class definition.

Auto answer

Auto Answer (Class ID: 28, version=1)				**P**	**M**	**Pr**
1	logical_name	Octet string	0-0:2.2.0.255		R	
2	Mode	enum	Value: (200) Manufactory specific Mode: CSD call is used to triggger GPRS connection.		R	
3	Listening_window	array	In case CSD wakeup is not used: Listening_window is never active.		RW	

			In case CSD wakeup is used: Listening window is always active			
4	Status	enum			R	
5	number_of_calls	unsigned	Default: (0) No limit		R	
6	number_of_rings	nr_rings_type	Default: Call is answered (rejected) after 1 ring. This means: nr_rings_in_window=1 nr_rings_out_of_window=1.		R	
7	list_of_allowed_callers	array[0]	empty		R	
8	list_of_callers_and_actions	array[32]	Array of callers with associated scripts. The script contains the push method invocation of the Push Setup CSD/SMS object		RW	
	Specific methods	***m/o***				

9.1.4 *External trigger: wakeup using SMS message*

The meter can be triggered by a SMS message to setup a GPRS connection. The listening window is always active in case SMS wakeup is used. The content of the SMS message is empty. If the calling number is in the list_of_allowed_senders_and_actions, the associated script is executed. This script is executed and the Push Setup object will trigger the connection management.

See annex C for the Class definition.

Message Handler

Message Handler (Class ID: 60, version=0)				**P**	**M**	**Pr**
1	logical_name	Octet string	0-0:2.3.0.255		R	
2	Listening_window	array	In case SMS wakeup is used: Listening_window is always active. In case SMS wakeup is not used: Listening window is never active.		RW	
3	list_of_allowed_senders	array	empty		R	
4	list_of_senders_and_actions	array	Array of senders with associated scripts. The script contains the push method invocation of the Push Setup CSD/SMS object.		RW	
	Specific methods	***m/o***				

9.2 Push script table

The Push script table holds scripts to activate the push operation. There are 3 entries in the script array. Every entry in the array of scripts calls the push method of one “Push setup” object instance.

Script table (Class ID: 9, version=0)				P	M	Pr
1	logical_name	Octet string	0-0:10.0.108.255		R	
2	Scripts	array[3]	Contains the script which contains the push method invocation of the Push Setup object n: Script 1 invokes the push method of Push Setup scheduled object. Script 2 invokes the push method of Push Setup Alarm object. Script 3 invokes the push method of Push Setup CSD/SMS object.		RW	
	Specific methods	***m/o***				
1	*Execute(data)*	***m***	Data contains the entry in the script table (1, 2 or 3)		X	

9.3 Push Setup

The push_object_list contains the information that is sent in the TCP message as mentioned in figure 2 of the GPRS Companion Standard.

If the push_object_list is empty, no TCP message is sent to the CS.

In the objectlist two objects are defined. The first object will point to an object that contains an octetstring. The second object points to the logical name of the triggering object (scheduler or SMS/CSD messagehandler) or the second object points to the value of the triggered object (register monitor). The objectlist values are sent as octet string separated by a comma (,).

Example 1:

(trigger from 0-0:15.0.4.255 and Equipment Id is XXXXX123456789012):

58,58,58,58,31,32,33,34,35,36,37,38,39,30,31,32,

2C,

00, 00, 0F, 00, 04, FF

Example 2:

(trigger from 0-0:16.1.0.255 and Equipment Id is XXXXX123456789012 and alarm register indicates a “replace battery alarm”):

58,58,58,58,31,32,33,34,35,36,37,38,39,30,31,32,

2C,

00, 00, 00, 02

The number of retries and repetitions_delay attributes determine if and when a retry is executed if the actions defined in the push object are not successful. This means that retries are started when one of the following actions were not successful:

- GPRS attach
- PDP context activation
- sending push information (if configured by the Push object via the push_object_list)
  - TCP client socket activation
  - sending message
  - receiving ACK
- TCP server socket activation.

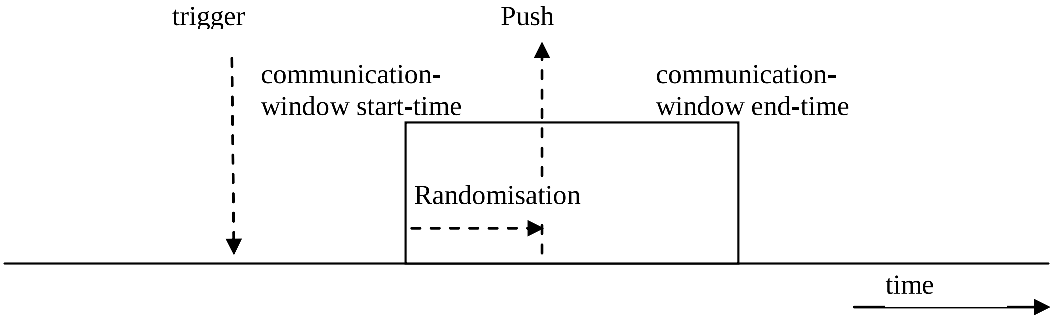
The repetitions and repetitions_delay attributes are used to guard the activation of PDP context and additionally guard the reception of an ACK of a send TCP message (if configured by the Push object via the push_object_list).

If multiple push objects are activated by multiple triggers at almost the same time then the first setup push method shall be finalized before the next push setup activation is executed. Finalized means in this case that communication on all the TCP sockets stopped and the meter closed PDP, TCP and GPRS connections and is ‘idle’ with regard to active communication sessions on P3.

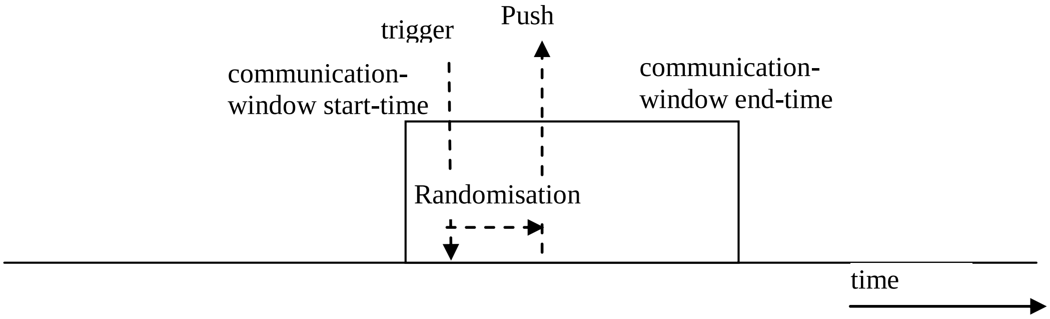
When a trigger for a Push action is received before the start of a communication window, the randomisation time will be determined from the start time of the window. See figure below. When the trigger for a Push action is received after the start-time of the communication window, then the randomisation will be determined from the start of the trigger. When the determined randomisation time falls outside the communication-window, then a new randomisation time will be determined in the next communication window.

The reason for this behaviour is that it shall be prevented that many meters will start a Push action at the same time.   In the above described way a power-up of many meters at the same time (after a power-down) during a communication-window will not result in many Push actions at the same time.

Trigger before communication-window:



Trigger within communication-window:



Push Setup Scheduler (Class ID: 40, version=0)				P	M	Pr
1	logical_name	Octet string	0-0:25.9.0.255		R	
2	push_object_list	array	This contains the TCP message that has to be sent: equipment ID and OBIS code of Push Scheduler {1,0-0:96.1.1.255,2,0 } {22,0-0:15.0.4.255,1,0} If the array is empty (array[0]), no TCP message is sent		RW	

3	send_destination_and_method	structure	Defines the destination of the TCP message that is optionally sent to the CS
			send_destination_and_method ::= structure
			{
			service service_type
			destination octet-string,
			message message_type
			}
			"service" is TCP, value 0
			"Destination" contains the destination address (DNS)
					RW	

			including port number "message" is 128 (i.e. first free manufacturer specific value: Octet String).			
4	communication_window	array	If the calling_window is empty (array[0]) this means the communication is possible without any limitations. If no starttime is specified then the communication will never start.		RW	
5	randomisation_start_interval	long-unsigned	in seconds (default value = 0). An interval of 0 means no randomisation; i.e. immediate start at the beginning of the first communication window		RW	
6	Number_of_retries	unsigned	The maximum number of retries in case of unsuccessful push attempts. A value of 0 means no repetitions		RW	
7	repetition_delay	long-unsigned	The time delay expressed in seconds until an unsuccessful push attempt can be repeated		RW	
	Specific methods	***m/o***				
1	push (data)	***m***			X	

Push Setup Alarm (Class ID: 40, version=0)				**P**	**M**	**Pr**
1	logical_name	Octet string	0-1:25.9.0.255		R	
2	push_object_list	array	This contains the TCP message that has to be sent: equipment ID and the value of the Alarm object. {1,0-0:96.1.1.255,2,0} {1,0-0:97.98.0.255,2,0} If the array is empty (array[0]), no TCP message is sent		RW	

3	send_destination_and_method	structure	Defines the destination of the TCP message that is optionally sent to the CS
			send_destination_and_method ::= structure
			{
			service service_type
			destination octet-string,
			message message_type
			}
			“service” is TCP, value 0
			“Destination” contains the destination address (DNS) including port number
			“message” is 128 (i.e. first free manufacturer specific value: Octet String).
			When “destination” is empty(= octet-string[0]) then the function is disabled.
					RW	

4	commiunication_window	array	If the calling_window is empty (array[0]) this means the communication is possible without any limitations. If no starttime is specified then the communication will never start.		RW	
5	randomisation_start_interval	long-unsigned	in seconds (default value = 0). An interval of 0 means no randomisation; i.e. immediate start at the beginning of the first communication window		RW	
6	number of retries	unsigned	The maximum number of retries in case of unsuccessful push attempts. A value of 0 means no repetitions		RW	
7	repetition_delay	long-unsigned	The time delay expressed in seconds until an unsuccessful push attempt can be repeated		RW	
	Specific methods	***m/o***				
1	push (data)	***m***			X	

Push Setup CSD/SMS (Class ID: 40, version=0)				P	M	Pr
1	logical_name	Octet string	0-2:25.9.0.255		R	

2	push_object_list	array	This contains the TCP message that has to be sent:
			equipment ID and OBIS code of CSD wakeup (auto answer) or
			{1,0-0:96.1.1.255,2,0}
			{28,0-0:2.2.0.255,1,0}
			or
			equipment ID and OBIS code of SMS wake (message handler):
			{1,0-0:96.1.1.255,2,0}
			{60,0-0:0.2.3.255,1,0}
			If the array is empty (array[0]), no TCP message is sent
					RW	

3	send_destination_and_method	structure	Defines the destination of the TCP message that is optionally sent to the CS
			send_destination_and_method ::= structure
			{
			service service_type
			destination octet-string,
			message message_type
			}
			“service” is TCP, value 0
			“Destination” contains the destination address (DNS) including port number
			“message” is 128 (i.e. first free manufacturer specific value: Octet String).
			When “destination” is empty(= octet-string[0]) then the function is disabled.
					RW	

4	communication_window	array	If the calling_window is empty (array[0]) this means the communication is possible without any limitations. If no starttime is specified then the communication will never start.		RW	
5	randomisation_start_interval	long-unsigned	in seconds (default value = 0). An interval of 0 means no randomisation; i.e. immediate start at the beginning of the first communication window		RW	
6	number_of_retries	unsigned	The maximum number of retries in case of unsuccessful push		RW	

			attempts. A value of 0 means no repetitions			
7	repetition_delay	long-unsigned	The time delay expressed in seconds until an unsuccessful push attempt can be repeated		RW	
	Specific methods	***m/o***				
1	push (data)	***m***			X	

Annex A :Number of digits (through the whole metering chain)

E meter		internally			DISPLAY normal mode			DISPLAY service mode			Remark
		register value	scaler (10^x)	unit	Fn(x,y)	value	unit	Fn(x,y)	value	unit	
Active energy import, rate 1 (+A)	1-0:1.8.1.255	123456789	0	Wh	F6(0,0)	123456	kWh	F9(3,3)	123456,789	kWh	
Active energy import, rate 2 (+A)	1-0:1.8.2.255	123456789	0	Wh	F6(0,0)	123456	kWh	F9(3,3)	123456,789	kWh	
Active energy export, rate 1 (-A)	1-0:2.8.1.255	123456789	0	Wh	F6(0,0)	123456	kWh	F9(3,3)	123456,789	kWh	
Active energy export, rate 2 (-A)	1-0:2.8.2.255	123456789	0	Wh	F6(0,0)	123456	kWh	F9(3,3)	123456,789	kWh	
Instantaneous active power import (+P)	1-0:1.7.0.255	123456	0	W	F5(3,3)	12,345	kW	n.r.	n.r.	kW	Novelle/AmvB requirement
Instantaneous active power export (-P)	1-0:2.7.0.255	123456	0	W	F5(3,3)	12,345	kW	n.r.	n.r.	kW	Novelle/AmvB requirement

G meter	M-Bus level G meter			DISPLAY normal mode			DISPLAY service mode			
	register value (DIF)	mulitplier (VIF)	unit	Fn(x,y)	value	unit	Fn(x,y)	value	unit	
Index value (temp converted)	8 digit BCD (most significant bits 12345678)	0,001	m³	F8(3,3) or F9(4,4)	12345,678*)	m³	F9(4,4)	12345,679	m³	Gas G4/G6 Ultrasonic and Diaphragm meters
Index value (temp converted)	8 digit BCD (most significant bits 12345678)	0,01	m³	F8(2,2) or F8(3,3)	123456,78*)	m³	F9(3,3)	123456,79	m³	Gas >G6 Ultrasonic and Diaphragm meters

For diaphragm meters with a mechanical display it is allowed to use a nonius for the last digit

*) One additional digit is allowed

P1: Communication with P1 device (IHD/OSM)		internally in E meter			P1 output				
		register value	scaler (10^x)	unit	Fn(x,y)	value	unit	comment	
Active energy import, rate 1 (+A)	1-0:1.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy import, rate 2 (+A)	1-0:1.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 1 (-A)	1-0:2.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 2 (-A)	1-0:2.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		

Instantaneous active power import (+P)	1-0:1.7.0.255	123456	0	W	F5(3,3)	12,345	kW		Novelle requirement
Instantaneous active power export (-P)	1-0:2.7.0.255	123456	0	W	F5(3,3)	12,345	kW		Novelle requirement
		register value (DIF)	mulitplier (VIF)	unit	Fn(x,y)	value	unit	comment	
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	0,001	m³	F8(3,3) or F9(4,4)	12345,678	m³		Gas G4/G6 Ultrasonic and Diaphragm meters
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	0,01	m³	F8(2,2) or F8(3,3)	123456,78	m³		Gas >G6 Ultrasonic and Diaphragm meters
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	10.000.000	J	Fn(2,2)	x,12	GJ	Heat/Cold	x: number of digits depending on the demands of the supplier of Heat/Cold
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	0,001	m³	Fn(3,3)	x,123	m³	Water	x: number of digits depending on the demands of the supplier of Water
M-bus Register n	0-n:24.2.1.255	12 digit BCD (most significant bits 123456789012)	0,001	Wh	F9(3,3)	123456,789	kWh	Slave E meter	

P2: M-bus communication level		**internally**							
		register value (DIF)	mulitplier (VIF)	unit	Fn(x,y)	representing	unit	comment	
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	0,001	m³	F8(3,3) or F9(4,4)	12345,678	m³		Gas G4/G6 Ultrasonic and Diaphragm meters

M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	0,01	m³	F8(2,2) or F8(3,3)	123456,78	m³		Gas >G6 Ultrasonic and Diaphragm meters
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	10.000.000	J	Fn(2,2)	x,12	GJ	Heat/Cold	x: number of digits depending on the demands of the supplier of Heat/Cold
M-bus Register n	0-n:24.2.1.255	8 digit BCD (most significant bits 12345678)	0,001	m³	Fn(3,3)	x,123	m³	Water	x: number of digits depending on the demands of the supplier of Water
M-bus Register n	0-n:24.2.1.255	12 digit BCD (most significant bits 123456789012)	0,001	Wh	F8(2,2)	123456,789	kWh	Slave E meter	

P3: Communication with CS		**internally**			**P3 output**				
		register value	scaler (10^x)	unit					
Active energy import (+A)	1-0:1.8.0.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy import, rate 1 (+A)	1-0:1.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy import, rate 2 (+A)	1-0:1.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export (-A)	1-0:2.8.0.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 1 (-A)	1-0:2.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 2 (-A)	1-0:2.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Instantaneous active power import (+P)	1-0:1.7.0.255	123456	0	W	F5(3,3)	123,45	kW		
Instantaneous active power export (-P)	1-0:2.7.0.255	123456	0	W	F5(3,3)	123,45	kW		
Profiles (hourly)	**0-x:24.3.0.255**	x=1,2,3 or 4 for each M-bus channel							
M-bus Register n	0-n:24.2.1.255					12345,678	m³		Gas G4/G6 Ultrasonic and Diaphragm meters
						123456,78	m³		Gas >G6 Ultrasonic and Diaphragm meters

						x,12	GJ		x: number of digits depending on the demands of the supplier of Heat/Cold
						x,123	m^3		x: number of digits depending on the demands of the supplier of Water
						123456,789	kWh	Slave E meter	
Load Profile period 1 (15 minutes)	**1-0:99.1.0.255**								
Active energy import (+A)	1-0:1.8.0.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export (-A)	1-0:2.8.0.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Load Profile period 2 (daily combined billing values)	**1-0:99.2.0.255**								
Active energy import, rate 1 (+A)	1-0:1.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy import, rate 2 (+A)	1-0:1.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 1 (-A)	1-0:2.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 2 (-A)	1-0:2.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
M-bus Register n	0-n:24.2.1.255					12345,678	m^3		Gas G4/G6 Ultrasonic and Diaphragm meters
						123456,78	m^3		Gas >G6 Ultrasonic and Diaphragm meters
						x,12	GJ		x: number of digits depending on the demands of the supplier of Heat/Cold
						x,123	m^3		x: number of digits depending on the demands of the supplier of Water
						123456,789	kWh	Slave E meter	
Billing Period 1 (monthly)	**0-0:98.1.0.255**								
Active energy import, rate 1 (+A)	1-0:1.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy import, rate 2 (+A)	1-0:1.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 1 (-A)	1-0:2.8.1.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
Active energy export, rate 2 (-A)	1-0:2.8.2.255	123456789	0	Wh	F9(3,3)	123456,789	kWh		
M-bus Register n	0-n:24.2.1.255					12345,678	m^3		Gas G4/G6 Ultrasonic and Diaphragm meters
						123456,78	m^3		Gas >G6 Ultrasonic and Diaphragm meters
						x,12	GJ		x: number of digits depending on the demands of

									the supplier of Heat/Cold
						x,123	m³		x: number of digits depending on the demands of the supplier of Water
						123456,789	kWh	Slave E meter	

Annex B: Standard Readout Object List

STANDARD READOUT OBJECT LISTS and Display Definitions		Auto scroll mode			Manual scroll mode			Service mode			During M-bus installation mode of E meter		P1 output
		Display in Auto scroll (5 seconds per Obis code)			Manual scroll of the display. Return to auto scroll after 30 seconds not pushing a button.			Manual scroll of the display. Only available in case of removed terminal cover. Return to auto scroll only after installed terminal cover.					Continuously when request line is activated
		General display read out Numeric row 0-0:21.0.1.255	**General display read out Alphanumeric row 0-0:21.0.1.255**	**Symbols / other parts of the display**	**Alternate display read out Numeric row 0-0:21.0.2.255**	**Alternate display read out Alphanumeric row 0-0:21.0.2.255**	**Symbols / other parts of the display**	**Service display read out Numeric row 0-0:21.0.3.255**	**Service display read out Alphanumeric row 0-0:21.0.3.255**	**Symbols / other parts of the display**	**Numeric row**	**Symbols / other parts of the display**	**General local port read out 0-0:21.0.0.255**
OBIS Code	**Description**												
n.a.	Display test	1	1	1	1	1	1						n.a.
n.a.	Manufacturer specific header												1
1-3:0.2.8.255	Version information				2								2
0-0:1.0.0.255	Actual date and time							1					3
0-0:96.1.1.255	Equipment identifier												4
1-0:1.8.1.255	Positive active energy Tariff 1 (A+)	2		+T01	3		+T01 and kWh symbol	2		+T01 and kWh symbol			5
1-0:1.8.2.255	Positive active energy Tariff 2 (A+)	3		+T02	4		+T02 and kWh symbol	3		+T02 and kWh symbol			6
1-0:2.8.1.255	Negative active energy Tariff 1 (A-)	4		-T01	5		-T01 and kWh symbol	4		-T01 and kWh symbol			7
1-0:2.8.2.255	Negative active energy Tariff 2 (A-)	5		-T02	6		-T02 and kWh symbol	5		-T02 and kWh symbol			8
0-0:96.14.0.255	Tariff indicator Electricity (actual tariff information)			T01 or T02			T01 or T02			T01 or T02			9
1-0:1.7.0.255	Instantaneous Active Power (A+, sum of all phases)												10
1-0:2.7.0.255	Instantaneous Active Power (A-, sum of all phases)												11
1-0:15.7.0.255	Instantaneous Active Power (abs(QI+QIV)-abs(QII+QIII)) (see note 5)		2,3,4,5	-P <- or -> +P and KW symbol		3,4,5,6	-P <- or -> +P and kW symbol		2,3,4,5	-P <- or -> +P and kW symbol			
0-1:24.1.0.255	Device type Channel 1 (formatted as text, eg. "Gas", "Water", "Warmte", "Koude", "Elektr.")					8a			7a				

0-1:24.1.0.255	Mbus Client Channel 1 serial number (see note 2)						8			7					
0-2:24.1.0.255	Device type Channel 2 (formatted as text)							9a			8a				
0-2:24.1.0.255	Mbus Client Channel 2 serial number (see note 2)						9			8					
0-3:24.1.0.255	Device type Channel 3 (formatted as text)							10a			9a				
0-3:24.1.0.255	Mbus Client Channel 3 serial number (see note 2)						10			9					
0-4:24.1.0.255	Device type Channel 4 (formatted as text)							11a			10a				
0-4:24.1.0.255	Mbus Client Channel 4 serial number (see note 2)						11			10					
1-0:0.2.0.255	Active Firmware identifier									11	0.0.2.0				
1-1:0.2.0.255	Module Active Firmware identifier									12	1.0.2.0				
1-0:21.7.0.255	Instantaneous Active Power L1 (A+)									13	0.21.7.0	kW symbol			
1-0:22.7.0.255	Instantaneous Active Power L1 (A-)									14	0.22.7.0	kW symbol			
1-0:41.7.0.255	Instantaneous Active Power L2 (A+) (see note 1)									15	0.41.7.0	kW symbol			
1-0:42.7.0.255	Instantaneous Active Power L2 (A-) (see note 1)									16	0.42.7.0	kW symbol			
1-0:61.7.0.255	Instantaneous Active Power L3 (A+) (see note 1)									17	0.61.7.0	kW symbol			
1-0:62.7.0.255	Instantaneous Active Power L3 (A-) (see note 1)									18	0.62.7.0	kW symbol			
n.a.	Active Threshold					LMT flag (see note 4)	7 (see note 4)		LMT flag (see note 4)		6 (see note 4)	LMT flag (see note 4)			
n.a.	Switch position electricity					closed breaker symbol			closed breaker symbol			closed breaker symbol			
0-0:96.7.21.255	Number of power failures in any phase														12
0-0:96.7.9.255	Number of long power failures in any phase														13
1-0:99.97.0.255	Power failure event log														14
1-0:32.32.0.255	Number of voltage sags in phase L1														15
1-0:52.32.0.255	Number of voltage sags in phase L2 (see note 1)														16
1-0:72.32.0.255	Number of voltage sags in phase L3 (see note 1)														17
1-0:32.36.0.255	Number of voltage swells in phase L1														18

1-0:52.36.0.255	Number of voltage swells in phase L2 (see note 1)														19
1-0:72.36.0.255	Number of voltage swells in phase L3 (see note 1)														20
0-0:96.13.1.255	Text message 8				6 (see note 6)										21
0-0:96.13.0.255	Text message 1024														22
0-1:24.1.0.255	M-Bus Client Channel 1 Device type (see note 2)														23
0-1:96.1.0.255	M-Bus Device ID 1 Channel 1 - Equipment Identifier (see note 2)														24
0-1:24.2.1.255	M-Bus Capture Time Channel 1														35
0-1:24.2.1.255	M-Bus Value 1 Channel 1														26
0-2:24.1.0.255	M-Bus Client Channel 2 Device type (see note 2)														27
0-2:96.1.0.255	M-Bus Device ID 1 Channel 2 - Equipment Identifier (see note 2)														28
0-2:24.2.1.255	M-Bus Capture Time Channel 2														29
0-2:24.2.1.255	M-bus Value 1 Channel 2														30
0-3:24.1.0.255	M-Bus Client Channel 3 Device type (see note 2)														31
0-3:96.1.0.255	M-Bus Device ID 1 Channel 3 - Equipment Identifier (see note 2)														32
0-3:24.2.1.255	M-Bus Capture Time Channel 3														33
0-3:24.2.1.255	M-bus Value 1 Channel 3														34
0-4:24.1.0.255	M-Bus Client Channel 4 Device type (see note 2)														35
0-4:96.1.0.255	M-Bus Device ID 1 Channel 4 - Equipment identifier (see note 2)														36
0-4:24.2.1.255	M-Bus Capture Time Channel 3														37
0-4:24.2.1.255	M-bus Value 1 Channel 3														38
1-0:32.7.0.255	Instantaneous voltage L1					L1			L1			L1 (see note 7)		L1 (see note 7)	
1-0:52.7.0.255	Instantaneous voltage L2 (see note 1)					L2			L2			L2 (see note 7)		L2 (see note 7)	
1-0:72.7.0.255	Instantaneous voltage L3 (see note 1)					L3			L3			L3 (see note 7)		L3 (see note 7)	
0-1:94.31.4.255	GPRS Network information					GPRS signal strength symbol			GPRS signal strength symbol			GPRS signal strength symbols			
0-1:94.31.0.255	Administrative In/Out status: Administrative Off ("Administratief Uit")					AU flag			AU flag			AU flag			

0-1:94.31.0.255	Administrative In/Out status: Default ("Standaard")					STD flag			STD flag			STD flag			
0-1:94.31.0.255	Administrative In/Out status: Administrative On ("Administratief In")					AI flag			AI flag			AI flag			
	Registered to GSM Netwerk					NW flag			NW flag			NW flag			
0-0:97.97.0.255	Error register (when at least one of the bits 0, 1, 2, or 3 from byte 2 is high)											FF flag			
	M-bus serial numbers (at least 7 digits) of all M-bus devices found												M-bus serial nr's in order (see note 8)		

Notes:

1) Polyphase meters only
2) Only if device is installed
3) Always off
4) Register value can either be positive or negative, the displayed value is always positive. Direction of power can be determined by the arrows of the actual power on the display
5) Visible untill button is pressed or empty message is sent from CS to meter
6) Blinking in case of registration of negative active energy as an indication of POSSIBLE wrong connection of "phase in" and "phase out" for that phase
7) Wireless M-bus devices: By pushing the button, the next serial number of a found M-bus device is shown. Selecting a device is done by pressing the button at least two seconds

   Wired M-bus devices: No selection necessary by means of a push button
8) In case the information is too large to show completely on the display, horizontal scrolling is allowed.

ANNEX C: FUTURE DLMS CHANGES

Greenbook Changes

Encryption of initiate request:

Current DLMS Greenbook section 9.3.2 text:

- When the dedicated key is present, the xDLMS InitiateRequest APDU shall be authenticated and encrypted using the AES-GCM-128 algorithm, the global unicast encryption key and the authentication key (if in use). The xDLMS InitiateRequest APDU shall be ciphered the same way, when the dedicated key is not present, but it is necessary to protect the RLRQ APDU by including the ciphered xDLMS InitiateRequest in its user-information field. See 9.3.3.

New DLMS Greenbook section 9.3.2 text, accepted in Budapest 11-07-08

- In the AARQ, the xDLMS-Initiate.request shall be authenticated and encrypted if the security policy is not (0) nothing or if the dedicated key is present. It may be authenticated and encrypted in any case
- In the RLRQ the the xDLMS-Initiate.request shall be always authenticated and encrypted if the application context is a ciphered one. *(Post meeting note: This assumes that the global keys are in place. Once they are transfered, they are immediatyle activated).*
- NOTE 1 The xDLMSInitiate.request is either not ciphered, or – when it is ciphered – it shall be authenticated and encrypted.
- NOTE 2 The xDLMS-Initiate.request can be ciphered in a ciphered application context only.
- NOTE 3: If the Security setup object is not present the request to establish an AA with ciphered context shall be refused *(Post meting note: This means that the Security setup object is mandatory and also the Frame counter object)*

Blue Book changes

Auto answer	0...n	class_id = 28, version = 1			
Attributes	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. mode (static)	enum				x + 0x08
3. listening_window (static)	array				x + 0x10
4. status (dyn.)	enum				x + 0x18
5. number_of_calls (static)	unsigned				x + 0x20
6. number_of_rings (static)	nr_rings_type				x + 0x28
7. list_of_allowed_callers (static)	array				x + 0x30
8. list_of_callers_and_actions (static)	array				x + 0x38
Specific methods	***m/o***				

Attribute description

logical_name Identifies the “Auto answer” object instance.

mode Defines the working mode of the line when the device is auto answering.

enum: (0) line dedicated to the device,
(1) shared line management with a limited number of calls allowed. Once the number of calls is reached, the window status becomes inactive until the next start date, whatever the result of the call,
(2) shared line management with a limited number of successful calls allowed. Once the number of successful communications is reached, the window status becomes inactive until the next start date,
(3) currently no modem connected,
(200...255) manufacturer specific modes

listening_window Contains the start and end instant when the window becomes active (for the start instant), and inactive (for the end instant). The start_time implicitly defines the period.

Example: when the day of month is not specified (equal to 0xFF) this means that we have a daily share line management. Daily, monthly …window management can be defined.

```
array     window_element
window_element ::= structure
{
          start_time:     octet-string,
          end_time:       octet-string
}
```

start_time and end_time are formatted as set in 4.1.6.1 for *date_time*

status Here the status of the window is defined.

enum:
(0) Inactive: the device will manage no new incoming call. This status is automatically reset to Active when the next listening window starts,
(1) Active: the device can answer to the next incoming call,
(2) Locked: This value can be set automatically by the device or by a specific client when this client has completed its reading session and wants to give the line back to the customer before the end of the window duration. This status is automatically reset to Active when the next listening window starts.

number_of_calls

This number is the reference used in modes 1 and 2.

When set to 0, this means there is no limit.

number_of_rings

Defines the number of rings before the meter connects the modem. Two cases are distinguished: number of rings within the window defined by attribute “listening_window” and number of rings outside the “listening_window”.

```
nr_rings_type ::= structure
{
    nr_rings_in_window:       unsigned, (0: no connect in window)
    nr_rings_out_of_window:   unsigned (0: no connect out of window)
}
```

list_of_allowed_callers

Contains a list of calling numbers for which the meter connects the modem. This requires the presence of a calling line identification (CLI).

list_of_allowed_callers ::= array caller_id

caller_id: octet-string

caller_id defines an allowed calling number. caller_id also supports wild-card characters '?' and '*'. With '?' any single character matches, with '*' any character string matches.
‘*’ can only be used at the beginning or at the end of a number, but neither in between nor alone.

Example 1: "+994193500" = only calls from "+994193500" are accepted.

Example 2: "+9941935????" = calls from all numbers in the range of "+99419350000" to "+99419359999" are accepted.

Example 3: "7777*" = calls from all numbers starting with "7777" are accepted.

Example 4: "*9000" = calls from all numbers ending with "9000" are accepted.

The check of the calling number is done after a successful evaluation of the other attributes, i.e. mode, listening_window and number_of_rings. If the calling number doesn't match an entry in the list, the modem is not connected.

If no numbers are defined at all (= empty array), the auto answer function is active without checking the caller (no CLI).

list_of_callers_and_actions	Contains the logical name of a “Script table” object and the script selector of the script to be executed for a calling number. If the calling number matches, the associated script is executed. array callers_and_actions_element
	callers_and_actions_element ::= structure
	{
	caller_id: octet-string,
	executed_script: script
	}
	caller_id: The same syntax applies as described in list_of_allowed_callers.
	script ::= structure
	{
	script_logical_name: octet-string,
	script_selector: long-unsigned
	}
	If the same calling number is defined in the list_of_callers_and_actions as well as in the list_of_allowed_callers, the script is only executed if the conditions to connect the modem haven't been met, e.g. the number_of_rings is not reached or the incoming call is outside the listening_window.
	If the calling number doesn't match any caller_id or if no callers and actions are defined at all (= empty array), no script is executed.

Wake-Up using Messages

The wake-up functionality using messages such as SMS, MMS, e-mail, etc. is modelled using a new "Message handler" class with class_id = 60. The instances of the "Message handler" objects use OBIS code "0-b:0.2.3.255".

This IC allows modelling how the device manages incoming messages such as SMS, MMS, e-mail, etc. as well as the execution of dedicated actions based on the identification of the sender. The message service used is implicitly defined by the modem used.

Message handler		**0...n**	**class_id = 60, version = 0**			
Attributes		***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name	(static)	octet-string				x
2. listening_window	(static)	array				x + 0x08
3. list_of_allowed_senders	(static)	array				x + 0x10
4. list_of_senders_and_actions	(static)	array				x + 0x18
Specific methods		***m/o***				

Attribute description

logical_name

Identifies the “Message handler” object instance.

listening_window

Contains the start and end instant when the listening window becomes active (for the start instant), and inactive (for the end instant). The start_time implicitly defines the period.

Example: when the day of month is not specified (equal to 0xFF) this means that we have a daily listening window.

```
array     window_element
window_element ::= structure
{
          start_time:      octet-string,
          end_time:        octet-string
}
```

start_time and end_time are formatted as set in 4.1.6.1 for *date_time*

list_of_allowed_ senders

Contains a list of senders for which the device accepts incoming messages. If the sender matches, the content of the message is processed. This requires the presence of a sender identification (e.g. calling line identification).

```
list_of_allowed_senders ::= array sender_id

sender_id:      octet-string
```

sender_id defines an allowed sender. sender_id also supports wild-card characters '?' and '*'. With '?' any single character matches, with '*' any character string matches.
‘*’ can only be used at the beginning or at the end but neither in between nor alone.

Example 1: "+994193500" = only messages from "+994193500" are accepted.

Example 2: "+9941935????" = all messages coming from a sender in the range of "+99419350000" to "+99419359999" are accepted.

Example 3: "7777*" = all messages coming from a sender starting with "7777" are accepted.

Example 4: "*9000" = all messages coming from a sender ending with "9000" are accepted.

The check of the sender_id is only done if the listening_window is currently active. If the sender doesn't match an entry in the list, the message is discarded. Empty messages are handled as defined in list_of_senders_and_actions.

If no sender_id's are defined at all (= empty array), the incoming messages are always accepted without checking its origin.

list_of_senders_ and_actions

Contains the logical name of a “Script table” object and the script selector of the script to be executed if an empty message is received from a matching sender.array senders_and_actions_element

senders_and_actions_element ::= structure

{
sender_id: octet-string,
executed_script: script
}

The same syntax applies as described in list_of_allowed_senders.

As soon as an empty message is received from a matching sender, the associated script is executed. The listening_window is ignored. If the message contains data, no script is executed and the message is discarded except if the sender is present in the list_of_allowed_senders.

If the message sender doesn't match any sender_id or if no callers and actions are defined at all (= empty array), the SMS is discarded.

Push Setup Class

The push setup class mainly contains a list of objects to be pushed as well as the push destination and the communication medium to be used. It also defines the communication time window and the handling of repetitions for a push operation. But please note that the initial trigger comes from somewhere else (e.g. from a scheduler, a monitor, a dedicated event, etc.).

Push setup	**0...n**	**class_id = 40, version = 0**			
Attribute(s)	***Data type***	***Min.***	***Max.***	***Def.***	**Short name**
1. logical_name (static)	octet-string				x
2. push_object_list (static)	array				x + 0x08
3. send_destination_and_method (static)	structure				x + 0x10
4. calling_window (static)	array				x + 0x18
5. randomisation_start_interval (static)	long-unsigned				x + 0x20
6. repetitions (static)	unsigned				x + 0x28
7. repetition_delay (static)	long-unsigned				x + 0x30
Specific methods	***m/o***				
1. push (data)	m				x + 0x38

Attribute description

push_object_list

Defines the list of attributes or objects to be pushed. Upon a call of the push (data) method the selected attributes are sent to the destination defined in send_destination_and_method.

array object_definition

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

- where attribute_index is a pointer to the attribute within the object. attribute_index 1 refers to the 1st attribute (i.e. the logical_name), attribute_index 2 to the 2nd, etc.); attribute_index 0 refers to all public attributes;
- where data_index is a pointer selecting a specific element of the attribute. The first element in the attribute structure is identified by data_index 1. If the attribute is not a structure, then the data_index has no meaning.
  If the attribute is the buffer of a profile, then the data_index can be used to define a selective access. Selective access is especially useful when reading data from very large arrays with entries ordered according to time, such as profiles. A group of entries within a time range (from ... to ...) as well as a subset of columns can de defined. It is possible to define recurring time ranges (e.g. the data of the last hour or the data of the previous day until now) using wildcards. Thus the selection is defined by coding data_index as follows:

data_index:	MS-Byte		LS-Byte
	Upper nibble	Lower nibble	

0x0000 = the whole buffer is pushed without selective access

0x0001 to 0x0FFF (0 – 61439) = normal data index used.

0x1000 to 0xFFFF = the buffer is pushed using selective access as defined by the MS-Byte and the LS-Byte below.

MS-Byte upper nibble: Used to define a selective access by index or time range.

0xF = last complete number of months
0xE = last complete number of days
0xD = last complete number of hours
0xC = last complete number of minutes
0xB = last number of seconds
0xA = last complete number of months up to now
0x9 = last complete number of days up to now
0x8 = last complete number of hours up to now
0x7 = last complete number of minutes up to now
0x6 ... 0x2 = reserved
0x1 = last number of entries
0x0 = normal data index used

	MS-Byte lower nibble: Defines the number of columns of the buffer which will be pushed. 0 = all columns 1 to 15 = number of columns LS-Byte: Used to define the quantity of the selected time range in case of selective access, i.e. the number of days, hours, month or the number of entries. Example 1: 0xE401 = the first 4 columns of the last complete day of the selected profile will be pushed. Example 2: 0x1080 = all columns of the last 128 entries of the profile will be pushed. NOTE 1 If the array is empty, the push operation is disabled. Nevertheless the IP connection is established. NOTE 2 The push_object_list attribute itself can be pushed as well to clearly identify the pushed data.

send_destination_and_method	Contains the destination address (e.g. phone number, email address, IP address) where the data specified by the push_object_list has to be sent, and the sending method relevant to the destination.
	send_destination_and_method ::= structure
	{
	service service_type
	destination octet-string,
	message message_type
	}
	where:service defines the type of service used to push the data
	service_type ::= enumerated
	(0) TCP
	(1) UDP
	(2) FTP
	(3) SMTP
	(4) SMS
	destination defines e.g. phone number, email address, IP address where the data has to be sent.
	If the destination is empty = octet-string[0], no communication channel will be established at all and the push will be suppressed.

	message_type identifies the encoding used. message_type::= enumerated (0) COSEM APDU (1) XML COSEM APDU (128...255) manufacturer specific all other values are reserved for future use

calling_window	Contains the start and end date/time stamp when the communication window(s) for the push become active (for the start instant), or inactive (for the end instant).
	array window_element
	window_element ::= structure
	{
	start_time: octet-string,
	end_time: octet-string
	}
	start_time and end_time are formatted as set in 4.1.6.1 for *date_time* including *w*ildcards.
	After a successful push the communication window is closed.
	At the end of a communication window an already started push is continued.
	If the calling_window is empty = array[0] this means that the communication is possible without any limitations.

randomisation_start_interval	To avoid simultaneous network connections of a lot of devices at exactly the same point in time, a randomisation interval in seconds can be defined. This means that the push operation is not started immediately at the beginning of the first communication window but started randomly delayed. An interval of 0 means no randomisation, i.e. the push is started immediately at the beginning of the first communication window.

	The value is not applied in case of repetitions, i.e. it is only active for the initial connection attempt. The randomisation_start_interval is applied to all communication windows, i.e. if the randomisation time is longer than the first window, the initial push attempt may be made during the first or any later window.
repetitions	The maximum number of retrials in case of unsuccessful push attempts. After a successful push no further push attempts are made until the push setup is triggered again. A value of 0 means no repetitions, i.e. only the initial connection attempt is made.
repetition_delay	The time delay, expressed in seconds until an unsuccessful push attempt can be repeated. A value of 0 immediately triggers a repetition without any delay. Note: The repetition delay is not controlled/influenced by the communication window. A repetition only can be made if a communication window is active at that time. Otherwise it is handled like an unsuccessful push attempt.
Method description	
push(data)	Activates the data push process leading to the elaboration and sending the data. data ::= integer(0)

Push Setup Instances

All push setup object instances use the OBIS code 0-0:25.7.e.255. The purpose and the use of any of these objects need to be defined in a companion specification.

Push Script Table

The push script table object is an instance of class 9 (script table) with OBIS code 0-b:10.0.108.

Normally every entry in the array of scripts calls the push method of one push setup instance.

Push Scheduler

For regular, time based triggering push scheduler objects are used. Every push scheduler object is an instance of class 22 (single action scheduler) with OBIS code 0-b:15.0.4.255. The executed_script attribute provides the link to the script in the push script table which then executes the push method of the desired push setup object.