



CMe3100 DLMS Plugin  
User's Manual  
v1.8

# CONTENTS

<b>1</b>	<b>DOCUMENT NOTES</b> .....	<b>4</b>
1.1	COPYRIGHT AND TRADEMARK .....	4
1.2	CONTACTS.....	4
<b>2</b>	<b>USING THIS MANUAL</b> .....	<b>5</b>
2.1	PURPOSE AND AUDIENCE .....	5
2.2	MODELS .....	5
2.3	ADDITIONAL AND UPDATED INFORMATION .....	5
<b>3</b>	<b>INTRODUCTION</b> .....	<b>6</b>
<b>4</b>	<b>INSTALLING THE PLUGIN</b> .....	<b>7</b>
4.1	VALIDATE LICENSE INFORMATION .....	7
4.2	INSTALLING THE PLUGIN.....	7
<b>5</b>	<b>PLUGIN SETTINGS</b> .....	<b>8</b>
5.1	OVERVIEW .....	8
5.2	DLMS SETTINGS .....	10
5.2.1	<i>Enable/disable the DLMS Plugin</i> .....	10
5.2.2	<i>Configure listening TCP port number</i> .....	10
5.2.3	<i>System title</i> .....	11
5.2.4	<i>Change meter addressing mode</i> .....	11
5.2.5	<i>Value interval for profiles</i> .....	11
5.2.6	<i>Security</i> .....	12
5.3	OBIS METER MAPPING .....	14
5.3.1	<i>Active meter mappings</i> .....	15
5.3.2	<i>Meter types in use</i> .....	16
5.3.3	<i>Create or modify a meter mapping file</i> .....	16
5.3.4	<i>Simplified mapping</i> .....	17
5.3.4.1	<i>Retrieving raw M-bus data</i> .....	18
5.3.4.2	<i>Retrieving decoded data</i> .....	18
5.3.5	<i>Excluding OBIS codes from Push Reports</i> .....	19
5.3.6	<i>Meter mapping file properties</i> .....	19
<b>6</b>	<b>CONFIGURE PUSH REPORTS</b> .....	<b>21</b>
6.1	DLMS PUSH REPORT SETTINGS .....	22
6.2	PUSH NOTIFICATIONS .....	22
6.3	DEFAULT DLMS SERVER SETTINGS.....	22
6.4	ENABLE PUSH REPORT OF LOAD PROFILE/BILLING PROFILE .....	23
6.4.1	<i>Select report type and schedule</i> .....	23
6.4.2	<i>Set DLMS server settings – report receiver</i> .....	24
6.5	ENABLE PUSH REPORT OF STANDARD EVENT LOG/FRAUD DETECTION LOG .....	24
<b>7</b>	<b>DLMS IMPLEMENTATION AND INTEGRATION</b> .....	<b>25</b>
7.1	SUPPORTED USE CASES .....	25
7.2	SESSION ESTABLISHMENT BETWEEN HES AND THE GATEWAY USING POLL OPERATIONS.....	25
7.2.1	<i>System title</i> .....	26
7.3	SECURITY .....	26
7.3.1	<i>No security (Lowest Level Security) authentication</i> .....	26
7.3.2	<i>Low Level Security (LLS) authentication</i> .....	26
7.3.3	<i>High Level Security (HLS) authentication</i> .....	27
7.3.4	<i>Password and key handling</i> .....	27
7.4	ADDRESSING OF LOGICAL DEVICES.....	27
7.4.1	<i>Logical device addressing</i> .....	28
7.4.2	<i>Channel selection</i> .....	28

7.5	QUERYING DATA FROM THE GATEWAY USING POLL OPERATION .....	28
7.5.1	<i>Management logical device COSEM objects</i> .....	28
7.5.2	<i>Meter logical device COSEM objects</i> .....	33
7.6	PUSH NOTIFICATIONS .....	39
7.6.1	<i>Periodical push data</i> .....	39
7.6.2	<i>Event triggered push of data</i> .....	42
<b>8</b>	<b>MANAGING CONFIGURATIONS .....</b>	<b>43</b>
8.1	CONFIGURATION .....	43
8.2	BACKUP .....	43
<b>9</b>	<b>DEVICE REPLACEMENT WORKFLOW .....</b>	<b>44</b>
<b>10</b>	<b>EXAMPLES AND GUIDES.....</b>	<b>45</b>
	<b>DOCUMENT HISTORY .....</b>	<b>46</b>
10.1	DOCUMENT SOFTWARE APPLIANCE .....	46
<b>11</b>	<b>REFERENCES .....</b>	<b>47</b>
<b>12</b>	<b>TERMS AND ABBREVIATIONS.....</b>	<b>48</b>
12.1	NUMBER REPRESENTATION.....	48

# 1 Document notes

All information in this manual, including product data, diagrams, charts, etc. represents information on products at the time of publication, and is subject to change without prior notice due to product improvements or other reasons. It is therefore recommended that customers contact Elvaco AB for the latest product information before purchasing a CMe Series product.

The documentation and product are provided on an “as is” basis only and may contain deficiencies or inadequacies. Elvaco AB takes no responsibility for damages, liabilities or other losses by using this product.

## 1.1 Copyright and Trademark

© 2022, Elvaco AB. All rights reserved. No part of the contents of this manual may be transmitted or reproduced in any form by any means without the written permission of Elvaco AB. Printed in Sweden.

CMe Series is a trademark of Elvaco AB, Sweden.

## 1.2 Contacts

Elvaco AB  
Kabelgatan 2T  
434 37 Kungsbacka  
SWEDEN  
Phone: +46 300 30250  
E-Mail: [info@elvaco.com](mailto:info@elvaco.com)

Elvaco AB Technical Support  
Phone: +46 300 434300  
E-Mail: [support@elvaco.se](mailto:support@elvaco.se)

Online: <https://www.elvaco.com>

## 2 Using this manual

### 2.1 Purpose and Audience

This document covers installation, configuration and usage of the CMe3100 DLMS Plugin add-on.

### 2.2 Models

CMe3100

### 2.3 Additional and updated information

Latest documentation version is available on Elvaco web site at <https://www.elvaco.com>

## 3 Introduction

The DLMS Plugin is one of the extensions (add-ons) available for the CMe3100 product. It can be ordered separately or pre-installed from the factory.

The DLMS Plugin extends the CMe3100 core services with support for DLMS over TCP/IP, providing a powerful and versatile suite of tools for Head-end System (HES) integration as well as the possibility to read and monitor any M-Bus meter using the DLMS protocol.

IEC 62056 is a set of standards for Electricity metering data exchange by International Electrotechnical Commission. The IEC 62056 standards are the International Standard versions of the DLMS/COSEM specification. DLMS or Device Language Message Specification (originally Distribution Line Message Specification), is the suite of standards developed and maintained by the DLMS User Association and has been adopted by the IEC TC13 WG14 into the IEC 62056 series of standards. The DLMS User Association maintains a D Type liaison with IEC TC13 WG14 responsible for international standards for meter data exchange and establishing the IEC 62056 series. In this role, the DLMS UA provides maintenance, registration and compliance certification services for IEC 62056 DLMS/COSEM.

## 4 Installing the Plugin

This section covers the installation procedure of the Plugin.

### 4.1 Validate license information

To be able to install the Plugin, the Product must have the license for *CMe3100 DLMS Plugin*. Validate the license information by navigating to *Device > Licenses & Add-ons*. See Figure 1 below.

The screenshot shows the 'Licenses & Add-ons' page. At the top, there is a navigation bar with tabs: Dashboards, Measurement series, Meters, Configuration, Device (selected), and Support li. Below the navigation bar, the page title is 'Licenses & Add-ons'. There are two tabs: Licenses (selected) and Add-ons. Below the tabs, there is a link 'Upload new license file'. The main content area is titled 'Current license(s)'. It contains a table with the following data:

Licenses	
Company	
Issued	2016-03-30
Subscription end date	2018-04-29
Valid through	Unlimited
Name	Description
CMe3100 Core	CMe3100 License for up to 256 devices
<b>CMe3100 DLMS Plugin</b>	<b>CMe3100 License for DLMS plugin</b>
CMe3100 JSON-RPC Plugin	CMe3100 License for JSON-RPC plugin
CMe3100 ModBus Plugin	CMe3100 License for ModBus plugin
CMe3100 REST Plugin	CMe3100 License for REST plugin

Figure 1 Licenses & Add-ons page

You may order licenses for any Add-ons through contacting sales at Elvaco AB.

## 5 Plugin settings

This section covers the configuration possibilities available in the DLMS Plugin.

DLMS Plugin settings may be viewed through either:

- Navigate to Configuration > Services > DLMS
- Click the *Settings Elvaco-DLMS* link in the Add-ons list to view the settings page of the DLMS Plugin.

Elvaco-DLMS	1.5.0-SNAPSHOT	2016-07-01	service	Yes (Uninstall now)	Yes Settings	Yes
					Elvaco-DLMS)	

Figure 2 DLMS Plugin Settings link

### 5.1 Overview

The default selected tab in the DLMS Plugin settings is the *Overview* page. The *Overview* page shows general information about the DLMS Plugin, see Figure 3.

System	Value
Enabled	true
Created	2016-07-01
Version	1.5.0-SNAPSHOT
Short description	DLMS service

Figure 3 DLMS Plugin overview



From the menu tab, you may access the other available settings pages:

- **Settings**
  - Page containing general DLMS configuration for the Plugin.
- **OBIS meter mapping**
  - Page containing settings for meter mapping information from M-Bus meter data to COSEM data objects with OBIS code notation.
- **Help**
  - Help section describing the DLMS Plugin, linking to this document.
- **End-user license agreement (EULA)**

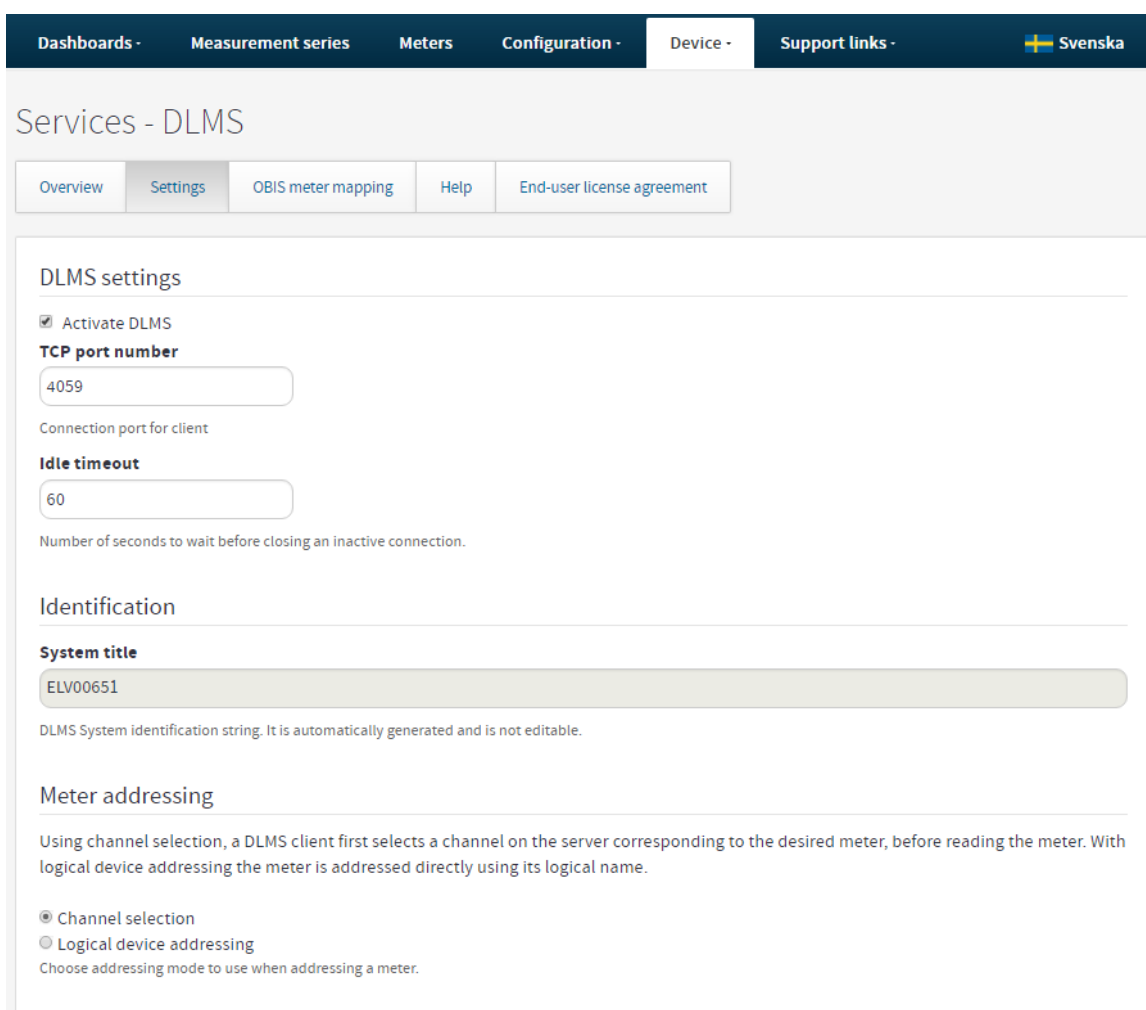
**Note! You need to create appropriate OBIS meter mappings for meters used in order for readout of historical data and Push Reports to work properly.**

## 5.2 DLMS Settings

This section covers the basic settings for the DLMS Plugin, see Figure 4. The settings page contains the following sections:

- DLMS Settings
- Value interval for profiles
- Security

Change settings to desired values and **click save to store changes**.



Services - DLMS

Overview Settings OBIS meter mapping Help End-user license agreement

### DLMS settings

Activate DLMS

**TCP port number**

4059

Connection port for client

**Idle timeout**

60

Number of seconds to wait before closing an inactive links.

### Identification

**System title**

ELV00651

DLMS System identification string. It is automatically generated and is not editable.

### Meter addressing

Using channel selection, a DLMS client first selects a channel on the server corresponding to the desired meter, before reading the meter. With logical device addressing the meter is addressed directly using its logical name.

Channel selection

Logical device addressing

Choose addressing mode to use when addressing a meter.

Figure 4 Settings page - DLMS Settings

### 5.2.1 Enable/disable the DLMS Plugin

Enable or disable the Plugin by clicking the *Activate DLMS* checkbox.

### 5.2.2 Configure listening TCP port number

The *TCP port number* is the listening port for which the Plugin will listen for incoming TCP connections. This port enable DLMS clients to access the on-board DLMS server of the Product.

The Plugin will automatically restart and start listening on the given port once changes are saved.

### 5.2.3 System title

The system title identifies the gateway during DLMS communication. This field is read-only and hold a per product unique string (shown for informational use only). See 7.2.1 for more information.

### 5.2.4 Change meter addressing mode

The Product enables two types of addressing modes; Channel selection and Logical device addressing, see section 7.4 for more information.

This parameter is no longer used. Both Channel selection and Logical device addressing are supported in parallel.

### 5.2.5 Value interval for profiles

The Product enables poll/push operation of three different generic profiles; Load profile 1, Load profile 2 and Billing profile.

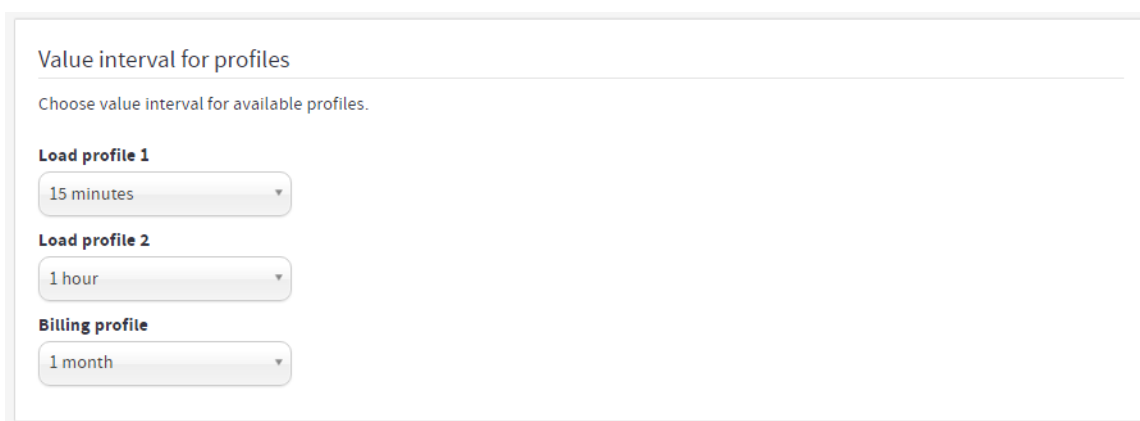


Figure 5 Settings page - Value interval for profiles

All three profiles can be configured individually to allow maximum flexibility regarding what value interval that is of interest.

**Please note that the value interval needs to be an even multiple of the meter readout schedule** (Meters > Readout schedule). For example, setting the meter readout schedule to 15 minutes and selecting 1 hour for load profile 1 will return every 4<sup>th</sup> value read from the meters.

Selecting "All values" will result in using the same value interval as configured for the meter readout schedule (see Figure 6 for a complete list of options).

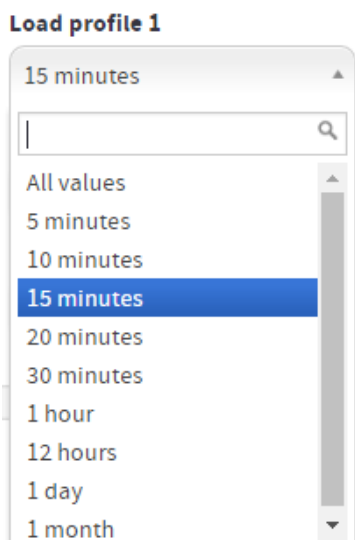


Figure 6 Value interval resolutions

### 5.2.6 Security

The Product supports **security suite 0** with none, low level or high level security. The security policy can also be changed using the DLMS protocol, see section 7.5.1.9. For security purposes the security policy can only be increased using the DLMS protocol. This means that if the security policy should be decreased, it must be done through using the GUI or any other configuration API for the Product.

For more information about security policy and which password and keys are used for authentication and encryption, see 7.3.

### Security

A client may connect using Low Level Security (LLS) or High Level Security (HLS). The security policy enforces what mechanisms that will be enforced, as a minimum, to authenticate the client and/or encrypt communication.

**Security policy**

None

Choose which security policy to enforce for connecting DLMS clients.

#### Low level security – Authentication

**Password**

.....

Password for Low Level Security (LLS) authentication.

#### High level security – Authentication & encryption (GMAC)

**Master Key / Key Encryption Key (KEK)**

.....

Pre-shared encryption key for client initiated update of Authentication key and Block cipher key (hex format, at least 16 characters).

**Authentication key**

.....

Key for HLS authentication (hex format, at least 16 characters).

**Block cipher key**

.....

Key for HLS encryption of communication (hex format, at least 16 characters).

Figure 7 Security

### 5.3 OBIS meter mapping

The OBIS meter mapping page is used to configure the mapping information needed to translate M-Bus meter values to readable COSEM objects.

The mapping table allows an M-Bus meter to be accessed using DLMS through mapping one or more M-Bus meter value fields to corresponding OBIS codes (and COSEM objects). Meter mappings can be created for a particular meter type, manufacturer and version or made generic for any meter of a specific type such as water, electricity etc.

**Note! You need to create appropriate OBIS meter mappings for meters used in order for readout of historical data and Push Reports to work properly.**

#### Recommended approach to meter mapping

Always strive for a common global mapping file that works for all meters in the system. This can be accomplished through for example:

- A generic mapping file for all meter of a particular type (water/heat/...) and, if needed, mapping several different M-Bus keys to the same OBIS code. The latter will bridge differences in how the M-Bus keys are used in different meters.
- M-Bus keys doesn't have to be present for all meters that are matched by the meter mapping file. This gives the possibility to use the same mapping file even if some meters don't support some of the M-Bus keys specified in the meter mapping file.

Dashboards -
Measurement series
Meters
Configuration -
Device -
Support links -
Svenska

## Services - DLMS

Overview
Settings
OBIS meter mapping
Help
End-user license agreement

### M-Bus to OBIS meter mapping

This mapping table allows a M-Bus meter to be accessed using DLMS through mapping one or more M-Bus meter value fields to corresponding OBIS codes (and COSEM objects). Meter mappings can be created for a particular meter type, manufacturer and version or made generic for any meter of a specific type such as water, electricity etc.

#### Active meter mappings

**Choose file to import**

Välj fil Ingen fil har valts

Add

M-Bus to OBIS meter mapping in a JSON formatted file. Download template from "Meter types in use" to add support for a new meter type.

Meter type	Manufacturer	Version	OBIS Codes	Options
water	DME	112	20/20	🗑️ ⚙️

Showing 1 to 1 of 1 entries

#### Meter types in use

To add support for a new meter type, use Options column to download a template for a specific meter and add OBIS mapping information. Meters shown are based on current meter list.

Meter type	Manufacturer	Version	Options
Heat	EFE	1	⬇️
Room sensor	ELV	30	⬇️
Heat	HYD	82	⬇️
Bus/System component	HYD	100	⬇️
Heat	KAM	25	⬇️
Warm water (30° C-90°C)	LUG	9	⬇️
Heat	LUG	4	⬇️

Showing 1 to 7 of 7 entries

First
Previous
1
Next
Last

Figure 8 OBIS meter mapping

### 5.3.1 Active meter mappings

The section *Active meter mappings* shows the current active meter mappings and for which meters they will be applied depending on if they match the meter type, manufacturer and version (based on M-Bus header information read from the meter). These fields can also be set to "all" if the meter mapping should be applied regardless of for example manufacturer or version.


The column *OBIS codes* shows the number of mapped M-Bus value fields to COSEM register objects. If there is no mapping for a specific M-Bus value, the DLMS Plugin will exclude that value in all poll/push operations and it will not be accessible.

To add a new meter mapping file or update an existing one:


- Click *Choose file* in the *Choose file to import* section and then click *Add*.

The updated or new meter mapping file will be available under *Active meter mapping* section. See section 5.3.3 for more information about how to create or modify meter mapping files.

To delete a meter mapping:

- Click the trashcan icon (  ) in the *Options* section.

### 5.3.2 Meter types in use

This section shows available/installed meters in the product. By clicking the download icon (  ), a template file is downloaded which serves as a good starting point for creating the OBIS meter mapping file. It contains all M-Bus fields available from the particular meter in the correct template format, ready to use as a meter mapping file.

**Please note that the CMe3100 must have done at least one readout from the meter before the template file is downloaded, otherwise the M-Bus value information will be empty.**

### 5.3.3 Create or modify a meter mapping file

To create a new meter mapping file, first use download for the appropriate meter as described in section 5.3.2. To modify an existing meter mapping file, download it in the section *Active meter mappings*.

The mapping files are in JSON format and can be edited in a text editor, preferably one that has support for highlighting JSON.

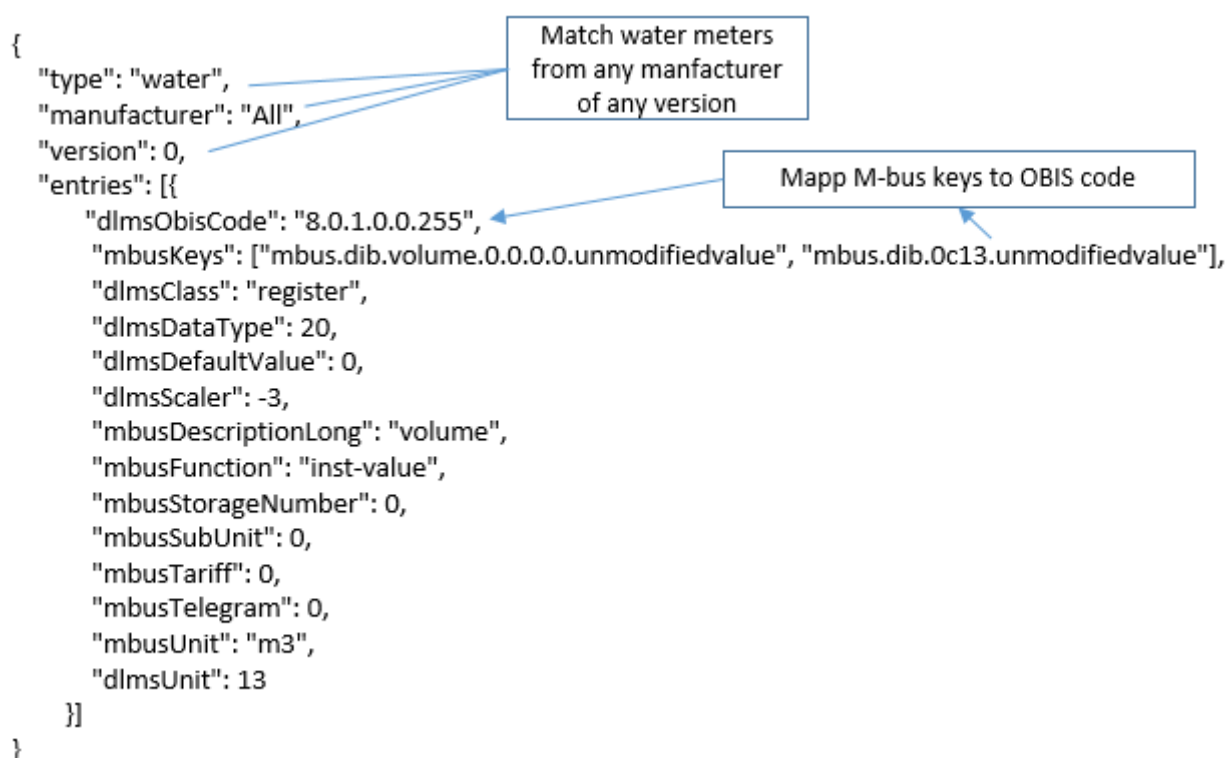
Update the contents of the file, by connecting OBIS codes with corresponding M-Bus keys (VIF/DIF combinations). All OBIS codes must be configured for each entry in the mapping, in order for the value to be available over DLMS.

The mapping can be made for a specific meter or made generic by setting manufacturer to “all” and/or version to 0. It is also possible to map several M-Bus keys, even from meters with different versions or manufacturers, to the same OBIS code. This allows you to create mapping files that handles for example meters which has differences between versions and/or manufacturers.

When using a generic mapping file it is also possible to combine it with other meter mapping files for specific manufacturers and/or a specific versions if they need to be handled in a different way. This is accomplished by the way the mapping files are applied, they use a predefined matching order as shown below. The first file that matches the criteria will be used. A meter mapping file which has manufacturer and/or version set to a specific value will be used instead of a generic mapping file if the manufacturer and/or version is matching for a particular meter.

1. Matching type, manufacturer and version.
2. Matching type, manufacturer and generic version.
3. Matching type, generic manufacturer and generic version.





The example file above matches all water meters and doesn't care about manufacturer or version. The translation entry will map both m-bus values with internal keys: "mbus.dib.volume.0.0.0.0.unmodifiedvalue" and "mbus.dib.0c13.unmodifiedvalue" to the same OBIS object: 8.0.1.0.0.255. If a new meter is added with a different DIF/VIF combination for the same value it could simply be supported by adding that new DIF/VIF to the list of "mbusKeys".

### 5.3.4 Simplified mapping

Simplified mapping allows all data to be queried over DLMS without mapping meter specific M-Bus keys (VIF/DIF combinations) to OBIS code.

There are two different ways to use simplified mapping:

1. Retrieving raw M-bus data.
2. Retrieving decoded M-bus data.

Raw M-bus data and decoded M-bus data can be retrieved over DLMS, by using the corresponding keys in table 10. This is useful when using meters from multiple manufacturers.

M-bus key	Suggested OBIS code	COSEM type	Description
mbus.frame.raw	0.0.1.1.128.255	Data / octet-string	Raw M-bus data
mbus.telegram.decoded.value	0.0.1.2.128.255	Data / visible-string	Decoded M-bus data
mbus.telegram.decoded.header	0.0.1.3.128.255	Data / visible-string	Headers for decoded M-bus data

Table 10b Simplified M-bus keys

Figure 8b shows custom keys for getting raw or decoded data.

### 5.3.4.1 Retrieving raw M-bus data

To retrieve raw M-bus data simply map an OBIS code to the “mbus.frame.raw” key. The M-bus data will be provided in the same way as if it was retrieved directly from the meter.

This is useful when the HES system handles decryption and decoding.

Below is a sample of how to map an OBIS code for getting raw M-bus data:

```

...
{
  "includeInPush": "ALL",
  "mbusTelegram": 0,
  "mbusKeys": ["mbus.frame.raw"],
  "dlmsObisCode": "0.0.1.1.128.255",
  "dlmsScaler": 0,
  "dlmsUnit": 0,
  "dlmsClass": "data",
  "dlmsDataType": 9,
  "dlmsDefaultValue": null,
  "mbusUnit": null,
  "mbusDescriptionLong": null,
  "mbusTariff": null,
  "mbusStorageNumber": 0,
  "mbusSubUnit": null,
  "mbusFunction": null
}
...

```

### 5.3.4.2 Retrieving decoded data

To retrieve decoded data over DLMS two OBIS codes must be used, one mapped to “mbus.telegram.decoded.value” for M-bus values and one mapped to “mbus.telegram.decoded.header” for header information.

It's important to always read both header and value information, as headers can change over time for a specific meter, for example as a result of a configuration change to the meter.

**Example header:**

```

#serial-number;device-identification;created;value-data-count;manufacturer;version;device-
type;access-number;status;signature;dif=0c,vif=78,description=fabrication-no,kind=inst-
value,tariff=0,subunit=0,storagenumber=0;dif=07,vif=7a,description=address,kind=inst-
value,tariff=0,subunit=0,storagenumber=0;dif=0c,vif=13,unit=m3,description=volume,kind=inst-
value,tariff=0,subunit=0,storagenumber=0;dif=8c10,vif=13,unit=m3,description=volume,kind=inst-
value,tariff=1,subunit=0,storagenumber=0;dif=0b,vif=3b,unit=m3/h,description=volume-
flow,kind=inst-value,tariff=0,subunit=0,storagenumber=0;dif=02,vif=5a,unit=°C,description=flow-
temp,kind=inst-value,tariff=0,subunit=0,storagenumber=0;dif=02,vif=fd70,description=datetime-
of-battery-change,kind=inst-
value,tariff=0,subunit=0,storagenumber=0;dif=0b,vif=26,unit=hour(s),description=op-
time,kind=inst-
value,tariff=0,subunit=0,storagenumber=0;dif=0a,vif=a618,unit=hour(s),description=op-time data-
error,kind=inst-
value,tariff=0,subunit=0,storagenumber=0;dif=04,vif=6d,description=datetime,kind=inst-
value,tariff=0,subunit=0,storagenumber=1

```

**Example values:**

```
0016000000;-2147483648;2000-01-01
01:00:00;00;HYD;37;water;10;0;0;47559461;47559461;85,946;0,000;0,000;27,300;14399;22525;
167;2016-01-06 16:50:00
```

**Mapping sample**

Below is a sample of how to map OBIS codes for getting decoded M-bus data:

```
...
{
  "includeInPush": "ALL",
  "mbusTelegram": 0,
  "mbusKeys": ["mbus.telegram.decoded.value "],
  "dlmsObisCode": "0.0.1.2.128.255",
  "dlmsScaler": 0,
  "dlmsUnit": 0,
  "dlmsClass": "data",
  "dlmsDataType": 10,
  "dlmsDefaultValue": null,
  "mbusUnit": null,
  "mbusDescriptionLong": null,
  "mbusTariff": null,
  "mbusStorageNumber": 0,
  "mbusSubUnit": null,
  "mbusFunction": null
}, {
  "includeInPush": "ALL",
  "mbusTelegram": 0,
  "mbusKeys": ["mbus.telegram.decoded.header "],
  "dlmsObisCode": "0.0.1.3.128.255",
  "dlmsScaler": 0,
  "dlmsUnit": 0,
  "dlmsClass": "data",
  "dlmsDataType": 10,
  "dlmsDefaultValue": null,
  "mbusUnit": null,
  "mbusDescriptionLong": null,
  "mbusTariff": null,
  "mbusStorageNumber": 0,
  "mbusSubUnit": null,
  "mbusFunction": null
},
...
```

### 5.3.5 Excluding OBIS codes from Push Reports

By default all OBIS codes included in a mapping file, will be included in all push reports for matching meters. It is possible remove a specific OBIS code by removing it from the mapping file, thus making it unavailable over DLMS, or setting the property “includeInPush” to “NONE”.

### 5.3.6 Meter mapping file properties

Top-level properties:

Property name	Type	Description
type	String	The type of meter that this translation applies to.

manufacturer	String	A three letter manufacturer code for which meter manufacturers this mapping applies to. Use <i>all</i> to match all manufacturers.
version	Integer	Numeric value for which meter version this mapping applies to. Use <i>0</i> to match all versions.
entries	JSON array	Translation entries defining the mapping between one or more M-Bus keys to a DLMS OBIS code.

Sub-properties for property *entries* (list of translation entries):

Property name	Type	Description
dlmsClass	String	COSEM type [clock, register, data, profile_generic]
dlmsDataType	Integer	DLMS type
dlmsDefaultValue	String	Default value used if a value is missing for the key. For no value use <i>null</i>
dlmsScaler	Integer	Numeric value used to scale the meters value. The value may be negative. This is the exponent (to the base of 10) of the multiplication factor.
dlmsUnit	Integer	Measure unit
mbusDescriptionLong	String	Not used.
mbusFunction	String	Not used.
includeInPush	String	Include this obis code in push reports, default is ALL [NONE, ALL]

## 6 Configure Push Reports

This section covers how to configure the DLMS Plugin Push Report support, which enables the CMe3100 to send DLMS Push Report to a HES with:

- Load Profile data
- Billing Profile data
- Standard Event Log (gateway and meters)
- Fraud Detection Log data (gateway and meters)
- Push Notification – Notify a DLMS client when the meter list has changed

Up to five different reports can be scheduled to run simultaneously. Each report has individual settings with regards to type of report (Load profile, Billing profile, Event log etc), schedule, server address and fallback server address.

Navigate to to *Configuration > Push Reports* to access the Push Report configuration page, see Figure 9.

Scheduled reports		Enabled
Report 1	Report is disabled	<input type="checkbox"/>
Report 2	Report is disabled	<input type="checkbox"/>
Report 3	Report is disabled	<input type="checkbox"/>
Report 4	Report is disabled	<input type="checkbox"/>
Report 5	Report is disabled	<input type="checkbox"/>

Event reports		Enabled
Boot-up report	Report is disabled	<input type="checkbox"/>
Installation report	Report is disabled	<input type="checkbox"/>
On demand report	Report is disabled	<input type="checkbox"/>

Figure 9 Push reports

## 6.1 DLMS Push Report settings

Navigate to *Configuration > Push Reports* and click the menu tab *DLMS Settings* to access the configuration page, see Figure 10.

Figure 10 Push Reports - DLMS settings

## 6.2 Push notifications

Enable or disable the Push notification when meter list changed by check/uncheck the checkbox. When enabled, the product will send an event to the default DLMS server when a meter is added, updated or removed.

See 7.6.2.1 for more information about the Push notifications.

## 6.3 Default DLMS server settings

Enter the default server address (IP or hostname) and TCP port to be used when sending push reports. The backup server address and TCP port is used as a fallback if a push report fails when trying to connect to the default server.

Enable or disable encrypted communication by check/uncheck the *Encrypt communication* checkbox.

## 6.4 Enable Push Report of Load Profile/Billing Profile

Click on the report (Report 1 to Report 5) in the *Push Reports* page to access the individual push report configuration page. Select *DLMS* in the drop-down field *Report type*, see Figure 11.

Figure 11 Report type DLMS

### 6.4.1 Select report type and schedule

Select report type 9010 (Load profile 1), 9011 (Load profile 2) or 9012 (Billing profile) in the drop-down field *Report template*.

Configure desired report schedule by selecting one of the items in the drop-down field *Report schedule*. The *Meter readout schedule* is read-only, and shows the current read out schedule for meters connected to the Product.

**Please note that the report schedule should be a multiple of the meter readout schedule.**

The report is sent as soon as the last meter readout cycle has completed. For example, setting the meter readout schedule to 15 minutes and selecting 1 hour for load profile 1, a report will be sent as soon as every 4<sup>th</sup> readout cycle completes.

In the *Report content* section (Figure 12), select the values to include in the report based on how long ago they were stored compared to when the report is sent. Use setting “Auto” to include all values not sent since last report was sent.

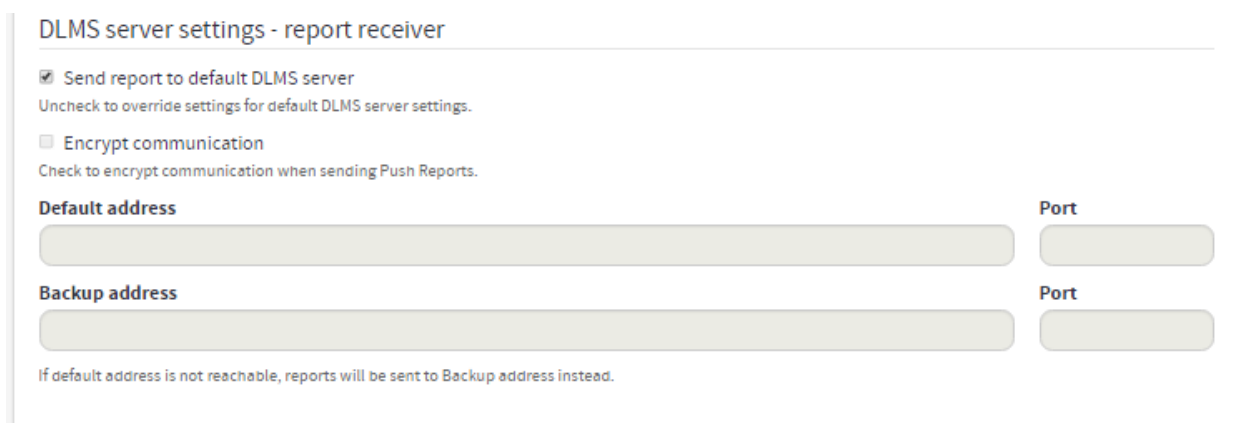
The *Value interval* is defined by the load profile/billing profile setting. See section 5.2.5 how to configure the individual load profile/billing profile value interval.

Figure 12 Report content

## 6.4.2 Set DLMS server settings – report receiver

Configure desired DLMS servers which should receive the Push Report in the *DLMS server settings – report receiver* section, see Figure 13.

Use the default server settings by check the *Send report to default DLMS server* checkbox. If the default server settings shouldn't be used, uncheck the checkbox and enter the desired DLMS server in *Default address* and *Port*. The *Backup address* and *Port* can be used as a fallback server if the report fails when sending report to the default address.



The screenshot shows a configuration window titled "DLMS server settings - report receiver". It contains the following elements:

- A checked checkbox labeled "Send report to default DLMS server" with the instruction "Uncheck to override settings for default DLMS server settings."
- An unchecked checkbox labeled "Encrypt communication" with the instruction "Check to encrypt communication when sending Push Reports."
- Two rows of input fields:
  - The first row is for "Default address" and "Port".
  - The second row is for "Backup address" and "Port".
- A note at the bottom: "If default address is not reachable, reports will be sent to Backup address instead."

Figure 13 DLMS server settings - report receivers

A test report can also be triggered by clicking the *Save and Send test report* button.

## 6.5 Enable Push Report of Standard event log/Fraud detection log

Standard event log and fraud detection log is configured in the same way as for load profile/billing profile, see section 6.4.

Select 9001 (Standard event log) or 9002 (Fraud detection log) in the *Report template* drop-down to enable desired log report.

Each report will push all events for both the gateway and the meters.



## 7 DLMS implementation and integration

The Product is tested according to the specification defined in the Yellow Book[3] for the DLMS certification.

The Product (physical device) comprises of the management logical device (Product) and one logical device for each connected meter. The logical devices can be addressed using channel selection or logical device addressing.

The management logical device holds COSEM management objects, i.e. a list of available logical devices, clock object, channel selection etc. The meter logical devices hold COSEM objects, which represent the meter data read from the connected meters.

The meter data is automatically decoded in the Product and a meter template is used to map the M-Bus data to COSEM objects with assigned OBIS-codes. The meter template can be configured by the end user.

Meter data can be read from the Product using poll operations or data can be sent to a server using push operations (configured using the Product GUI). The push operations can be enabled to send load profiles, billing profiles, event log data and event notifications.

### 7.1 Supported use cases

Please see below for an overview of supported use cases:

- **Polling operations using TCP/IP and xDLMS Services**
  - Detection of logical devices using SAP Assignment
  - Detection of COSEM objects using association LN (Long Names)
  - Read load and billing profiles of connected M-Bus meters
  - Read event logs
  - Configure security for none, low level security and high level security mode 5 (GMAC)
  - Logical device addressing or channel selection
- **Push operations using TCP/IP using Push Notification Services**
  - Up to 5 different push setups with:
    - Individually configurable schedules
    - Automatic retries and resending of unsent report data (e.g. due to network connectivity problems)
    - Jittered timestamp for transmission (load balancing receiving servers)
    - Fall-back server address
  - Push of load and billing profile
  - Push of event logs (standard events and fraud detection)
  - Push event "On Installation" – notification when meter list changed

### 7.2 Session establishment between HES and the Gateway using poll operations

The HES connects to the Gateway using TCP/IP. The HES associates with the Gateway on the DLMS/COSEM application layer using the following security features:

- Security suite 0
- Authentication: None, Low Level Security or High Level Security mode 5 (see [1] 9.2.74)
- Security policy: "None" or "Authenticated and encrypted" (see [1] 9.2.7.2.3)
- Server System title according to IDIS (see [4] 6.1.1 and section 7.2.1)
- Client SAP: 1 (management client) or 16 (public client)
- Server SAP: 1 (management logical device), 16..n (meter logical device)

### 7.2.1 System title

The system title is 8 bytes long and is coded according to IDIS (see [4] 6.1.1):

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8
MC	MC	MC	T1 <sub>b</sub>	T2 <sub>b</sub>	SN <sub>b</sub>	SN <sub>b</sub>	SN <sub>b</sub>

**MC:** Manufacturer Code according to FLAG Association as ASCII characters

**T1<sub>b</sub>:** IDIS Meter Device Type (Always set to 0)

**T2<sub>b</sub>:** IDIS Meter Function Type (Always set to 0)

**SN<sub>b</sub>:** Manufacturer specific serial number (28 bits, low nibble of byte 5 and byte 6..byte 8). Taken from the gateway serial number and represented as a binary 28 bit integer with MSByte order, right justified to byte 8.

The system title is built up from the logical device name of management logical device name.

**Example:**

**Management logical device name (gateway logical device name):**

ASCII characters: ELV000016000000

String hex representation: 0x454C56303030303136303030303030

**System title:**

Byte array hex representation: 0x454C560000F42400

## 7.3 Security

There are three different authentication mechanisms in DLMS; No security authentication, Low Level Security (LLS) authentication and High Level Security (HLS) authentication.

The authentication level is specified by the DLMS client when connecting to the Product.

### 7.3.1 No security (Lowest Level Security) authentication

The purpose of No security (Lowest Level Security) authentication is to allow the client to retrieve some basic information from the server. Typically, this is topology information including the name and the association view of the device.

This security mechanism is accepted only for the public client (id 16).

### 7.3.2 Low Level Security (LLS) authentication

When using LLS authentication the server requires that the client authenticates itself by supplying a password that is known by the server.

### 7.3.3 High Level Security (HLS) authentication

When running HLS authentication both the client and the server have to successfully authenticate themselves to establish a connection (in DLMS known as Application Association or AA). It is a four-pass process and there are several HLS authentication mechanisms available, e.g. GMAC. HLS requires a block cypher key to encrypt and decrypt messages sent between client and server.

For additional security an authentication key denoted AK is also specified. DLMS/COSEM supports something called key exchange which is a process of securely change encryption keys. When doing such an exchange the Key Encrypting Key (KEK) is used to wrap the keys before sending them. KEK is also known as the DLMS Master Key.

See Table 1 for more information about the keys and their usage.

Key	Description
Master Key / Key Encryption Key (KEK)	<p>A key encrypting key (KEK) is used to encrypt/decrypt other symmetric keys. In DLMS/COSEM this is the master key. KEK is used by DLMS client and server when exchanging keys.</p> <p>The key must be at least 16 characters long.</p>
Authentication key	<p>In DLMS, for additional security, an authentication key denoted AK is also specified. When present, it shall be part of the Additional Authenticated Data, AAD.</p> <p>The key must be at least 16 characters long.</p>
Block cipher key	<p>The block cipher key, also known as Encryption Key (EK), is used in the AES-GCM algorithm.</p> <p>The key must be at least 16 characters long.</p>

Table 1 Encryption keys description

### 7.3.4 Password and key handling

The password described in section 7.3.3 is by default set to *12345678*. If this key is used for authentication, it's highly recommended to change the key after installing the DLMS plugin.

All keys described in section 7.3.3 will be automatically generated the first time the DLMS plugin starts.

The keys can be updated in two ways. One way to update them is to change them on the DLMS settings page as described in section 5.2.6.

The other way is to have them updated by the HES as described in section 7.3.

## 7.4 Addressing of logical devices

The logical devices can be addressed using the logical device address or a manufacturer implemented channel selection mechanism. The channel and the logical device address are mapped one to one (same numeric value).

The manufacturer specific channel selection mechanism is implemented to reduce the handshaking overhead and enables the HES to access all logical devices in one single association.

### 7.4.1 Logical device addressing

The management logical device can be addressed using logical device address 1. Connected M-Bus meters (meter logical devices) are addressed using logical device address 16..n. A gateway with a single M-Bus meter connected will consequently have two accessible logical devices; the management logical device at address 1 and the meter logical device at address 16. If more meters are connected, the next free logical device address will be used, i.e. 17.

Logical device addressing can only be performed during session establishment.

### 7.4.2 Channel selection

Channel selection is available after successful association with any of the logical devices.

Channel selection can be used to address and access the different logical devices in the physical device (gateway). The manufacturer specific OBIS code 0.128.1.0.0.255 has been assigned for this purpose. The COSEM object is of interface class Data (1). The object holds a single LongUnsigned integer, which is readable and writable on all logical devices.

The default channel selected after a successful association is the same as the logical device address.

The HES sets the value of this object using xDLMS SET Service. Following APDU (XML formatted for readability) shows the format of such request selecting the management logical device:

```
<SetRequest>
  <SetRequestNormal>
    <InvokeIdAndPriority Value="C1" />
    <AttributeDescriptor>
      <ClassId Value="0001" />
      <InstanceId Value="0080010000FF" />
      <AttributeId Value="02" />
    </AttributeDescriptor>
    <Value>
      <Long-Unsigned Value="01" />
    </Value>
  </SetRequestNormal>
</SetRequest>
```

## 7.5 Querying data from the gateway using poll operation

After successful association, the logical device to address can be changed using channel selection. The gateway will stay on the current selected logical device until disconnect or a new channel selection is made during the same session.

Depending on selected logical device and the client id, different COSEM objects are available.

### 7.5.1 Management logical device COSEM objects

Following COSEM objects are available in the management logical device:

Description	OBIS Code	Interface Class	Access
-------------	-----------	-----------------	--------

	A	B	C	D	E	F		Public client (16)	Management client (1)
SAP Assignment	0	0	41	0	0	255	SAP Assignment (17)	R	R
Association LN	0	0	40	0	0	255	Association LN (15)	R	RW
Clock	0	0	1	0	0	255	Clock (8)	R	RW
Logical Device Name	0	0	42	0	0	255	Data (1)	R	R
Receive Frame Counter	0	0	43	1	0	255	Data (1)	R	R
Channel Selection	0	128	1	0	0	255	Data (1)	None	RW
Meter List	1	128	0	0	0	255	Data (1)	None	R
Gateway System Title	0	0	96	1	0	255	Data (1)	None	R
Gateway Firmware Version	1	0	0	2	0	255	Data (1)	None	R
Gateway Firmware Signature	1	1	0	2	8	255	Data (1)	None	R
Gateway IP Address	0	0	96	128	0	255	Data (1)	None	R
Standard Event Log - Event Id	0	0	96	11	0	255	Data (1)	None	R
Standard Event Log - Event Message	0	128	96	11	0	255	Data (1)	None	R
Fraud Event Log - Event Id	0	0	96	11	1	255	Data (1)	None	R
Fraud Event Log - Event Message	0	128	96	11	1	255	Data (1)	None	R
Specific Test Object	0	0	96	1	4	255	Data (1)	None	RW
Standard Event Log	0	0	99	98	0	255	Profile Generic (7)	None	R
Fraud Event Log	0	0	99	98	1	255	Profile Generic (7)	None	R
Push Setup Gateway Standard Event Log	0	4	25	9	0	255	Push Setup (40)	None	R
Push Setup Gateway Fraud Event Log	0	8	25	9	0	255	Push Setup (40)	None	R
Push Setup On Install	0	7	25	9	0	255	Push Setup (40)	None	R
Security Setup	0	0	43	0	0	255	Security Setup (64)	None	RW

Table 2 Management logical device COSEM Objects

### 7.5.1.1

#### 7.5.1.1.1 SAP Assignment

SAP Assignment object can be read from the gateway to identify available logical devices.

#### 7.5.1.2 Association LN

This object holds the list of available COSEM objects on current selected logical device.

This object can also be used by the HES to set the LLS authentication password.

#### 7.5.1.3 Clock

The clock object enables read/write of the internal gateway real time clock.

#### 7.5.1.4 Logical Device Name

Data type: octet-string

This object holds the logical device name of the gateway, 13 characters octet string.

The format of the logical device name is according to the following pattern:

**ELV<serial number>** Example: ELV0016000001

### 7.5.1.5 Receive Frame Counter

Data type: double-long-unsigned

This object holds the current value of the frame counter (invocation counter) needed for encrypted communication.

The gateway will drop any message carrying a frame counter with the value less than the gateway current value.

### 7.5.1.6 Channel Selection

Data type: long-unsigned

This object enables switching between all logical devices during a single association.

See 7.4.2 for more information.

### 7.5.1.7 Meter list

The meter list contains a list of meters, representing the meter logical device name and the corresponding channel, which can be used for channel selection.

The APDU (XML formatted for readability) of the meter list should have the following format:

```
<structure>
  <octet-string>COSEM logical device name of the gateway</octet-string>
  <array>
    <structure>
      <octet-string>COSEM logical device name of meter 1</octet-string>
      <long-unsigned>Channel of meter 1</long-unsigned>
    </structure>
    <structure>
      ...
    </structure>
    <structure>
      <octet-string>COSEM logical device name of meter n</octet-string>
      <long-unsigned>Channel of meter n</long-unsigned>
    </structure>
  </array>
</structure>
```

### 7.5.1.8 Gateway System Title

Data type: octet-string

This object holds the system title of the gateway See 7.2.1 for more information.

### 7.5.1.9 Gateway Firmware Version

Data type: octet-string

This object holds the gateway firmware version information.

### 7.5.1.10 Gateway Firmware Signature

Data type: octet-string

This object holds the gateway firmware signature information.

### 7.5.1.11 Gateway IP Address

Data type: double-long-unsigned

This object holds the gateway IPv4 address numeric representation.

For example, the address 192.168.1.2 is represented as 0xC0A80102, decimal value 3232235778.

### 7.5.1.12 Standard Event Log – Event Id

Data type: long-unsigned

This object holds the event id of the last event logged to Standard Event Log.

### 7.5.1.13 Standard Event Log – Event Message

Data type: visible-string

This object holds the event description of the last event logged to Standard Event Log.

### 7.5.1.14 Fraud Event Log – Event Id

Data type: long-unsigned

This object holds the event id of the last event logged to Fraud Event Log.

### 7.5.1.15 Fraud Event Log – Event Message

Data type: visible-string

This object holds the event description of the last event logged to Fraud Event Log.

### 7.5.1.16 Specific Test Object

Data type: octet-string

This object is used for testing and it should be ignored. It doesn't hold any information.

### 7.5.1.17 Standard event log and fraud detection log

The standard event log is a table of log items of type Profile Generic with gateway related log items. The data which can be read from the log is date/time of event and event code.

#### Attributes:

1: Logical name

2: Buffer (actual data)

Selector 0 (all data) and selector 1(from-to) implemented

3: Capture objects (list of OBIS codes)

4: N/A

5: N/A

6: N/A

7: Entries in use in the database

8: Same as 7

**COSEM objects/OBIS codes:**

Date/Time: 0.0.1.0.0.255

Event object: 0.0.96.11.0.255 (Standard), 0.0.96.11.1.255 (Fraud)

Event message: 0.128.96.11.0.255 (Standard), 0.128.96.11.1.255 (Fraud)

**Limitations:**

Maximum of 1000 rows can be read in one request.

See [6] for available log items.

**7.5.1.18 Push Setup Gateway Standard Event Log**

The push setup object holds the information about the objects to be included in the Standard Event Log Push report. For more information see 7.6.1.2 Push Event logs

**Attributes:**

1: Logical name

2: Push Object List

3: N/A

4: N/A

5: N/A

6: N/A

7: N/A

**7.5.1.19 Push Setup Gateway Fraud Event Log**

The push setup object holds the information about the objects to be included in the Fraud Event Log Push report. For more information see 7.6.1.2 Push Event logs

**Attributes:**

1: Logical name

2: Push Object List

3: N/A

4: N/A

5: N/A

6: N/A

7: N/A

**7.5.1.20 Push Setup On Installation**

The push setup object holds the information about the objects to be included in the On Installation Event Push report. For more information see 7.6.2.1 Push On-Installation Event



**Attributes:**

- 1: Logical name
- 2: Push Object List
- 3: N/A
- 4: N/A
- 5: N/A
- 6: N/A
- 7: N/A

**7.5.1.21 Security setup**

The security setup object enables read/write of keys and policy used in security suite 0.

For the key exchange, the HES calls methods of the security setup (version 0), OBIS code 0.0.43.0.0.255. The only keys that are to be exchanged are those of the gateway. There are no separate keys for the meters. The key exchange is to be executed as specified by [1] 9.2.5.4 and [4] 9.4.6.1.

When writing to the security setup object, the new security configuration will **NOT** be used until a disconnect and a new connection is made.

**7.5.2 Meter logical device COSEM objects**

Following COSEM objects are available in each meter logical device:

Description	OBIS Code						Interface Class	Access	
	A	B	C	D	E	F		Public client (16)	Management client (1)
SAP Assignment	0	0	41	0	0	255	SAP Assignment (17)	R	R
Association LN	0	0	40	0	0	255	Association LN (15)	R	R
Clock	0	0	1	0	0	255	Clock (8)	R	RW
Logical Device Name	0	0	42	0	0	255	Data (1)	R	R
Receive Frame Counter	0	0	43	1	0	255	Data (1)	R	R
Channel Selection	0	128	1	0	0	255	Data (1)	None	RW
Gateway Logical Device Name	1	128	2	0	0	255	Data (1)	None	R
Meter Firmware Version	1	0	0	2	8	255	Data (1)	None	R
Meter Manufacturer Id	0	0	96	128	1	255	Data (1)	None	R
Meter Device Type	0	0	96	128	2	255	Data (1)	None	R
Standard Event Log - Event Id	0	0	96	11	2	255	Data (1)	None	R
Standard Event Log - Event Message	0	128	96	11	2	255	Data (1)	None	R
Fraud Event Log - Event Id	0	0	96	11	3	255	Data (1)	None	R
Fraud Event Log - Event Message	0	128	96	11	3	255	Data (1)	None	R
Specific Test Object	0	0	96	1	4	255	Data (1)	None	RW

Load Profile 1	8	0	99	1	0	255	Profile Generic (7)	None	R
Load Profile 1	8	0	99	2	0	255	Profile Generic (7)	None	R
Billing Data	8	0	98	1	0	255	Profile Generic (7)	None	R
Standard Event Log	8	0	99	98	2	255	Profile Generic (7)	None	R
Fraud Event Log	8	0	99	98	3	255	Profile Generic (7)	None	R
Push Setup Meter Load Profile 1	0	1	25	9	0	255	Push Setup (40)	None	R
Push Setup Meter Load Profile 2	0	2	25	9	0	255	Push Setup (40)	None	R
Push Setup Meter Billing Data	0	3	25	9	0	255	Push Setup (40)	None	R
Push Setup Meter Standard Event Log	0	4	25	9	0	255	Push Setup (40)	None	R
Push Setup Meter Fraud Event Log	0	8	25	9	0	255	Push Setup (40)	None	R
Security Setup	0	0	43	0	0	255	Security Setup (64)	None	R
M-Bus registers dynamically created from available data in the M-Bus meter representing the meter logical device X=According to OBIS meter mapping, see section 5.3.	X	X	X	X	X	X	Register (3)	None	R
M-Bus data dynamically created from available data in the M-Bus meter representing the meter logical device X=According to OBIS meter mapping, see section 5.3.	X	X	X	X	X	X	Data (1)	None	R

Table 3 Meter logical device COSEM objects

### 7.5.2.1 SAP Assignment

SAP Assignment object can be read from the gateway to identify available logical devices

### 7.5.2.2 Association LN

This object holds the list of available COSEM objects on current selected logical device.

The COSEM objects for a specific meter logical device is a reflection of the available data registers read from an M-Bus meter which has valid mapping information in the OBIS meter mapping and the static objects according to Table 3. See section 5.3 for more information about the OBIS meter mapping.

### 7.5.2.3 Clock

The clock object enables read/write of the internal gateway real time clock.

### 7.5.2.4 Logical Device Name

Data type: octet-string

This object holds the logical device name of the meter, 13 characters octet string.

The format of the logical device name is according to the following pattern:

**<man><type><version><serial number>** Example: ELV010112345678

Where

<man> = Three capital letter manufacturer number according to Flag Association representing the meter manufacturer.

<type> = Two digit ASCII hex (one byte) type/medium of meter according to [4] section 5.8 representing the meter medium.

<version> = Two digits ASCII hex (one byte) version of the meter.

#### 7.5.2.5 Receive Frame Counter

Data type: double-long-unsigned

This object holds the current value of the frame counter (invocation counter) needed for encrypted communication.

The gateway will drop any message carrying a frame counter with the value less than the gateway current value.

#### 7.5.2.6 Channel Selection

Data type: long-unsigned

This object enables switching between all logical devices during a single association.

See 7.4.2 for more information.

#### 7.5.2.7 Gateway Logical device name

This object is a mirror of the logical device name of the gateway and is accessible on all meter logical devices in the gateway. This enables read of the gateway logical device name when associating/addressing a meter logical device. See 7.5.1.4 for more information.

#### 7.5.2.8 Meter Firmware Version

Data type: octet-string

This object holds the meter firmware version information.

#### 7.5.2.9 Meter Manufacturer Id

Data type: octet-string

This object holds the meter manufacturer id information.

#### 7.5.2.10 Meter Device Type

Data type: integer

This object holds the meter MBus device type information.

#### 7.5.2.11 Standard Event Log – Event Id

Data type: long-unsigned

This object holds the event id of the last event logged to Standard Event Log.

#### 7.5.2.12 Standard Event Log – Event Message

Data type: visible-string

This object holds the event description of the last event logged to Standard Event Log.

### 7.5.2.13 Fraud Event Log – Event Id

Data type: long-unsigned

This object holds the event id of the last event logged to Fraud Event Log.

### 7.5.2.14 Fraud Event Log – Event Message

Data type: visible-string

This object holds the event description of the last event logged to Fraud Event Log.

### 7.5.2.15 Specific Test Object

Data type: octet-string

This object is used for testing and it should be ignored. It doesn't hold any information.

### 7.5.2.16 Billing profile and Load profile 1 & 2

The billing profile is a table with all registers read from the M-Bus meter which contains valid mapping in the OBIS meter mapping. The billing profile resolution is configurable according to section 5.2.5.

#### Attributes:

- 1: Logical name
- 2: Buffer (actual data)
  - Selector 0 (all data) and selector 1(from-to) implemented
- 3: Capture objects (list of OBIS codes)
- 4: N/A
- 5: N/A
- 6: N/A
- 7: Entries in use in the database
- 8: Same as 7

#### COSEM objects/OBIS codes:

Date/Time: 0.0.1.0.0.255

M-Bus registers: Depends on the OBIS meter mapping file (see section 5.3)

#### Limitations:

Maximum of 1000 rows can be read in one request.

### 7.5.2.17 Standard event log and fraud detection log

The standard event log is a table of log items of type Profile Generic with meter related log items. The data which can be read from the log is date/time of event and event code. The event code is the status byte read from the M-Bus meter + base value (4000). This gives the possibility to get the actual M-Bus status code readable from the event code value (4000..4255). Log items will only be created in the log if the M-Bus status byte changes between readout.

The fraud detection log contains a table of log items of type Profile Generic with meter related log items. The data which can be read from the log is date/time of event and event code.

Currently there are no fraud detection log items in the meter fraud detection log, i.e. reading the log will return zero (0) rows.

**Attributes:**

- 1: Logical name
- 2: Buffer (actual data)
  - Selector 0 (all data) and selector 1(from-to) implemented
- 3: Capture objects (list of OBIS codes)
- 4: N/A
- 5: N/A
- 6: N/A
- 7: Entries in use in the database
- 8: Same as 7

**COSEM objects/OBIS codes:**

- Date/Time: 0.0.1.0.0.255
- Event object: 0.0.96.11.0.255 (Standard), 0.0.96.11.1.255 (Fraud)
- Event message: 0.128.96.11.0.255 (Standard), 0.128.96.11.1.255 (Fraud)

**Limitations:**

Maximum of 1000 rows can be read in one request.

See [6] for available log items.

**7.5.2.18 Push Setup Meter Load Profile 1**

The push setup object holds the information about the objects to be included in the Load Profile 1 Push report. For more information see 7.6.1.1 Push Profiles

**Attributes:**

- 1: Logical name
- 2: Push Object List
- 3: N/A
- 4: N/A
- 5: N/A
- 6: N/A
- 7: N/A

### 7.5.2.19 Push Setup Meter Load Profile 2

The push setup object holds the information about the objects to be included in the Load Profile 2 Push report. For more information see 7.6.1.1 Push Profiles

**Attributes:**

- 1: Logical name
- 2: Push Object List
- 3: N/A
- 4: N/A
- 5: N/A
- 6: N/A
- 7: N/A

### 7.5.2.20 Push Setup Meter Billing Data

The push setup object holds the information about the objects to be included in the Billing Data Push report. For more information see 7.6.1.1 Push Profiles

**Attributes:**

- 1: Logical name
- 2: Push Object List
- 3: N/A
- 4: N/A
- 5: N/A
- 6: N/A
- 7: N/A

### 7.5.2.21 Push Setup Gateway Standard Event Log

The push setup object holds the information about the objects to be included in the Standard Event Log Push report. For more information see 7.6.1.2 Push Event logs

**Attributes:**

- 1: Logical name
- 2: Push Object List
- 3: N/A
- 4: N/A
- 5: N/A
- 6: N/A
- 7: N/A

### 7.5.2.22 Push Setup Gateway Fraud Event Log

The push setup object holds the information about the objects to be included in the Fraud Event Log Push report. For more information see 7.6.1.2 Push Event logs

**Attributes:**

---

- 1: Logical name
- 2: Push Object List
- 3: N/A
- 4: N/A
- 5: N/A
- 6: N/A
- 7: N/A

### 7.5.2.23 Security setup

The security setup object enables reading of the security parameters.

The keys and policy can be changed only in the gateway instance of the security setup object.

### 7.5.2.24 M-Bus data and registers

Available M-Bus registers are mapped into COSEM objects according to the OBIS meter mapping. The OBIS meter mapping templates can be configured by the end user according to the guidelines in section 5.3.

All M-Bus registers are of type interface class Register and contains the attributes *value*, *unit* and *scaler*. The translation from M-Bus register data to COSEM object tries to keep the original data type as far as possible, which means that integers and threatened as integers, floats as floats etc.

In order to minimize the load on the meters the DLMS plugin has a caching mechanism for M-Bus registers.

When reading M-Bus registers the DLMS plugin will first look in the cache for the value. If it exists, and the timestamp of the value is less than 60 seconds old, the value is returned to the caller.

If the value is older or does not exist in the cache, a readout from the meter is done. This operation will interrupt any readout cycles in case they are running.

If the readout is successful, the value is written to cache and returned to the caller. In the case of a failing meter readout, the plugin will return the last known value, if it exists in the cache, otherwise a DLMS error message is generated.

Reading a COSEM object (included in a meter template), which contains no valid data, will return the default error value defined in the meter template.

## 7.6 Push notifications

The gateway supports to periodically push data or send data on events. Push notifications are sent using Data-Notification-Service.

### 7.6.1 Periodical push data

Periodical push data is configured according to section 6.

#### 7.6.1.1 Push Profiles

Profiles are pushed according to configured schedule (section 6) via the Data-Notification-Service for all connected meters. A push contains data from one meter. It may contain data from multiple profiles. The payload of the Data Notification APDU has the following structure:

```
<octet-string>COSEM logical device name of the gateway</octet-string>
```

```

<octet-string>COSEM logical device name of the meter</octet-string>
<octet-string>COSEM logical name of the push setup object</octet-string>
<octet-string>COSEM logical name of profile</octet-string>
<array>
  <structure>capture object 1</structure>
  ...
  <structure>capture object m</structure>
</array>
<array>
  <structure>profile buffer entry 1</structure>
  ...
  <structure>profile buffer entry n</structure>
</array>
<array>
  <structure>unit and scaler for register 1 stored in profile</structure>
  ...
  <structure>unit and scaler for register m stored in profile</structure>
</array>

```

**Each capture object must contain the following elements:**

```

<long-unsigned>interface class ID</long-unsigned>
<octet-string>COSEM logical name</octet-string>
<integer>attribute index</integer>
<long-unsigned>data index</long-unsigned>

```

(see [2] 4.3.6)

**Each profile buffer entry must contain the following elements:**

```

<octet-string>timestamp</octet-string>
<some datatype>value</some data type>
<some datatype>value</some data type>
<some datatype>value</some data type>
...

```

(see [2] 4.3.6)

**Each unit and scaler object contains the following elements:**

```

<integer>scaler</integer>
<enum>unit</enum>

```

(see [2] 4.3.2)



### 7.6.1.2 Push Event logs

Event logs (gateway and meters) are pushed according to configured schedule (see section 6.1) via the Data-Notification Service.

A push contains data from a single logical device (gateway or meter). It may contain data from multiple event logs. Standard event log is available for both meter logical device and gateway logical device. The fraud detection log is only available for gateway logical device as the actual meaning of the meter event (which may be manufacturer specific) is not decoded by the gateway. Hence all meter events are placed in the "Meter status" event section available through the standard event log.

See CMe3100 Event logs [6] for available log items. The standard event log contains all events but those in the "Security" event section. The fraud detection log *only* contains events from the "Security" event section.

**The payload of the Data Notification APDU has the following structure for Gateway logical device (management device):**

```

<octet-string>COSEM logical device name of the gateway</octet-string>
<octet-string>COSEM logical name of the push setup object</octet-string>
<octet-string>COSEM logical name of event log</octet-string>
<array>
  <structure>event 1</structure>
  ...
  <structure>event n</structure>
</array>
  
```

**The payload of the Data Notification APDU has the following structure for a meter logical device:**

```

<octet-string>COSEM logical device name of the gateway</octet-string>
<octet-string>COSEM logical device name of the meter</octet-string>
<octet-string>COSEM logical name of the push setup object</octet-string>
<octet-string>COSEM logical name of event log</octet-string>
<octet-string>manufacturer ID</octet-string>
<integer>device type</integer>
<integer>firmware version</integer>
<array>
  <structure>event 1</structure>
  ...
  <structure>event n</structure>
</array>
  
```

**Each event must contain the following items:**

```
<octet-string>timestamp of the event</octet-string>  
<octet-string>OBIS code of the event object</octet-string>  
<unsigned>event code</unsigned>
```

**Optionally, it may also contain another item:**

```
<visible-string>custom event message</visible-string>
```

**OBIS codes used:**

Timestamp: 0.0.1.0.0.255

Event object: 0.0.96.11.0.255 (Standard), 0.0.96.11.1.255 (Fraud)

Custom event message: 0.128.96.11.0.255 (Standard), 0.128.96.11.1.255 (Fraud)

## 7.6.2 Event triggered push of data

### 7.6.2.1 Push On-Installation Event

This event is triggered when the meter list of a gateway is changed. The event triggers a push with the event code On-Installation, which can be used by the HES to trigger a poll operation to update the meter list (read meter list from the gateway).

The gateway will only trigger a single push, even if more than one meter is added/removed during an operation. The hysteresis for grouping the event is three (3) minutes.

The payload of the Data Notification APDU has the following structure:

```
<octet-string>COSEM logical device name of the gateway</octet-string>  
<octet-string>COSEM logical name of the push setup - on installation  
object</octet-string>  
<double-long-unsigned>ip adress of the gateway</double-long-unsigned>  
<octet-string> System title of the gateway as defined in [4] 6.1.1</octet-  
string>
```

The OBIS code for the push setup – on installation object is fixed to 0-7:25.9.0\*255 (see [4], section 7.2.1).

## 8 Managing configurations

The product feature for managing configurations and backups also include settings for DLMS Plugin (see “Manage configurations” section of the Product’s user interface).

As configuration files are used for duplicating settings between many devices the DLMS password and keys are *not* part of the Configuration file. They are however included in the Backup file (see section 8.2).

### 8.1 Configuration

The configuration contains all properties that can be edited in the Product. These can be used to duplicate configuration to several devices since they do not contain any device specific settings.

To save a copy of the current configuration:

- Navigate to Configuration > Manage configurations.
- Select Configuration as File type.
- Click Execute.

When operation has finished the File repository section will contain the newly created configuration file. This can then be used to update a device and/or revert erroneous configurations.

To import a saved configuration:

- Navigate to Configuration > Manage configurations.
- Click Choose File and select the exported configuration file from the old device.
- Click upload.

To revert a configuration:

- Navigate to Configuration > Manage configurations.
- Click on the reload icon in the Action column of the configuration to revert to.

### 8.2 Backup

The backup contains Configuration as described in 8.1 as well as a copy of the Products databases and operation system settings.

This backup contains device specific settings and is therefore not suitable for replicating configuration to another device, it should only be used to restore a device’s settings after a physical replacement.

To make a backup:

- Navigate to Configuration > Manage configurations.
- Select Backup as File type.
- Click Execute.

When operation has finished the File repository section will contain the newly created backup file.

## 9 Device replacement workflow

In the case a device needs to be replaced the configuration should be backed up from the device being replaced and import to the replacement device. The steps for accomplishing this are described in section 8.

*Please note that the system title of the replacement device will be new since it's derived from the devices serial number. The HES might need to be updated accordingly.*

## 10 Examples and guides

Please contact Elvaco to obtain integration support and access to example code (Java/DLMS).

Available examples and guides:

- Reading historical data for load profile 1, load profile 2, billing period
- On-demand readout of a meter
- Creating or updating an OBIS meter mapping file

## Document History

Version	Date	Description	Author
1.0	2016-06-13	Initial version	David Vonasek
1.1	2016-07-20	Added description of System title in section 7.2.1.	David Vonasek
1.2	2016-09-06	Added description of password and key handling in section 7.3.4. Added section 8 (Managing configurations). Added section 9 (Device replacement workflow)	Peter Janson
1.3	2016-09-16	Clarified content of standard event log and fraud detection log.	Emil Tirén
1.4	2016-11-21	Added new field in meter log push message containing gateway logical name (section 7.6.1.2).	Peter Janson
1.5	2017-01-23	Added more verbose description about OBIS meter mapping.	Emil Tirén
1.6	2017-03-30	Added 5.3.4-5.3.4.2 describing simplified mapping	Daniel Svensson
1.7	2021-11-09	Add description of the new objects Update OBIS codes Fix description of the Push Profiles structure Document review	Tobias Svensson
1.8	2022-02-23	Updates in graphical design	Ericha Bloom

### 10.1 Document software appliance

Type	FW Version	Comments
CMe3100	>= 1.9.2	

## 11 References

- [1] DLMS UA 1000-2 Ed. 8.0 (Green book)
- [2] DLMS UA 1000-1 Ed. 12.0 (Blue book)
- [3] DLMS UA 1001-1 Ed. 5.0 (Yellow book)
- [4] IDIS Package 2 IP profile Ed 2.0
- [5] EN13757-3:2013, Communication systems for and remote reading of meters – Part 3: Dedication application layer
- [6] CMe3100 Log Events

## 12 Terms and Abbreviations

Abbreviation	Description
Plugin	CMe3100 DLMS Plugin
Product	CMe3100
GW	Gateway (CMe3100)
HES	Head-end system
COSEM	Companion Specification for Energy Metering
DLMS	Device language Message Specification
OBIS	Object Identification System
IDIS	Interoperable Device Interface Specifications

### 12.1 Number representation

Decimal numbers are represented as normal number, i.e. 10 (ten).

Hexadecimal numbers are represented with prefix 0x, i.e. 0x0A (ten)

Binary numbers are represented with prefix 0b, i.e. 0b00001010 (ten)