DLMS/COSEM Interface Classes and Data Structures Reference

Table of Contents

Introduction

This document provides a comprehensive reference for the DLMS/COSEM Interface Classes (ICs) and data structures implemented in our smart meter simulator. It follows the DLMS/COSEM standard specifications.

OBIS Codes

OBIS Code Structure

OBIS codes are 6-byte identifiers with the format: A-B:C.D.E.F

- A: Media (0=abstract, 1=electricity, 6=heat, 7=gas)
- B: Channel (0=no channel, 1-64=channel number)
- C: Physical value (current, voltage, energy)
- D: Measurement type (total, rate 1, rate 2)
- E: Tariff (0=total, 1-63=tariff rate)
- F: Billing period (0=current, 1-63=historical values)

Common OBIS Codes

```
Clock:                  0.0.1.0.0.255
Active Power +:         1.0.1.7.0.255
Active Power -:         1.0.2.7.0.255
Reactive Power +:       1.0.3.7.0.255
Reactive Power -:       1.0.4.7.0.255
Active Energy +:        1.0.1.8.0.255
Active Energy -:        1.0.2.8.0.255
Reactive Energy +:      1.0.3.8.0.255
Reactive Energy -:      1.0.4.8.0.255
Voltage L1:            1.0.32.7.0.255
Voltage L2:            1.0.52.7.0.255
Voltage L3:            1.0.72.7.0.255
Current L1:            1.0.31.7.0.255
Current L2:            1.0.51.7.0.255
Current L3:            1.0.71.7.0.255
```

Interface Classes

1. Data (IC: 1)

Basic class for storing simple data values.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. value (any type, read-write)

2. Register (IC: 3)

Stores numeric values with scaling and units.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. value (numeric, read-write)
    3. scaler_unit (structure, read-only)
    4. status (unsigned, read-write)
- Methods:
    1. reset()

3. Extended Register (IC: 4)

Extends Register with capture time.

- Additional Attributes:
    5. capture_time (date-time, read-write)
    6. status_valid (boolean, read-only)
- Additional Methods:
    2. capture()

4. Clock (IC: 8)

Manages device time and calendar.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. time (date-time, read-write)
    3. time_zone (int16, read-write)
    4. status (unsigned8, read-write)
    5. daylight_savings_begin (date-time, read-write)
    6. daylight_savings_end (date-time, read-write)
    7. daylight_savings_deviation (int8, read-write)
    8. daylight_savings_enabled (boolean, read-write)
    9. clock_base (enum, read-only)

5. Script Table (IC: 9)

Stores and executes scripts.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. scripts (array, read-write)
- Methods:
    1. execute_script()

6. Activity Calendar (IC: 20)

Manages tariff schedules and seasons.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. calendar_name (string, read-write)
    3. season_profile (array, read-write)
    4. week_profile_table (array, read-write)
    5. day_profile_table (array, read-write)
    6. active_season_profile (array, read-only)
    7. active_week_profile_table (array, read-only)
    8. active_day_profile_table (array, read-only)

7. Association LN (IC: 15)

Manages logical name referencing association.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. object_list (array, read-only)
    3. associated_partners_id (structure, read-only)
    4. application_context_name (octet-string, read-only)
    5. xdlms_context_info (structure, read-only)
    6. authentication_mechanism_name (octet-string, read-only)
    7. secret (octet-string, read-write)
    8. association_status (enum, read-only)
    9. security_setup_reference (octet-string, read-only)
- Methods:
    1. reply_to_HLS_authentication()
    2. change_HLS_secret()

8. Security Setup (IC: 64)

Manages security features and settings.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. security_policy (enum, read-write)
    3. security_suite (enum, read-write)
    4. client_system_title (octet-string, read-only)
    5. server_system_title (octet-string, read-only)
- Methods:
    1. security_activate()
    2. key_transfer()
    3. key_agreement()
    4. generate_key_pair()

9. Push Setup (IC: 40)

Configures and manages push operations.

- Attributes:
    1. logical_name (octet-string, read-only)
    2. push_object_list (array, read-write)
    3. send_destination_and_method (structure, read-write)
    4. communication_window (array, read-write)

5. randomisation_start_interval (unsigned16, read-write)
6. number_of_retries (unsigned8, read-write)
7. repetition_delay (unsigned16, read-write)

- Methods:

1. push()

Data Types

Basic Types

- null-data
- boolean
- bit-string
- double-long
- double-long-unsigned
- octet-string
- visible-string
- utf8-string
- bcd
- integer
- long
- unsigned
- long-unsigned
- long64
- long64-unsigned
- enum
- float32
- float64
- date-time
- date
- time

Complex Types

- array
- structure
- compact-array

Access Rights

- no-access
- read-only
- write-only
- read-write

Security

Security Levels

```
enum SecurityLevel {
    NONE = 0,
    LOW = 1,
    HIGH = 2,
    HIGH_GMAC = 3,
    HIGH_SHA256 = 4,
    HIGH_ECDSA = 5
}
```

Authentication Types

```
enum AuthenticationType {
    NONE = 0,
    LOW = 1,
    HIGH = 2,
    HIGH_MD5 = 3,
    HIGH_SHA1 = 4,
    HIGH_GMAC = 5,
    HIGH_SHA256 = 6,
    HIGH_ECDSA = 7
}
```

Security Policies

1. Nothing (No security)
2. All messages authenticated
3. All messages encrypted
4. All messages authenticated and encrypted

Security Suites

- AES-GCM-128 for encryption
- GMAC for authentication
- ECDH for key agreement

Key Management

- Master key
- Authentication key
- Encryption key
- Key transfer and update mechanisms